The Design of Compact Attitude Determination Control Systems

a project presented to

The Faculty of the Department of Aerospace Engineering

San José State University

In partial fulfillment of the requirements for the degree

Master of Science in Aerospace Engineering

By

Khiem Huu Gia Nguyen

June 2025

approved by

Dr. Periklis Papadopoulos

Faculty Advisor

ABSTRACT

The Design of Compact 0.5U Altitude Determination Control Systems

Khiem Huu Gia Nguyen

The current stat of the art of small satellite must meet high pointing requirements while operating under high constraint size, weight and power limitations. This paper's purpose is to present a possible introduction of consumer electronics into creating end-to-end design hardware implementation and preliminary performance of a compact 0.5U Ventura ADCS module that fit inside of TechEdSat.

The tetrahedral quadrant of brushless DC motor is integrated with two MEMS IMUs , fluxgate magnetometer and 6 sun sensors laying out on 6 faces of the CubeSat. The electronics are distributed over a four-layer 95 mm x 89 mm printed circuit board assembly. The connectors interface used are Pico-Lock connectors as well as 28 AWG PTFE wire harness for space grade resiliency.

The hardware in the loop models was created in MATLAB/Simulink with the implantation of 3 Degrees of Freedom simulation of the dynamics of spacecraft IMU modeling, disturbances modeling and sensor fusion modeling. The hardware in the loop test bench shows that the module can deliver 1.2 mN*m peak torque and 2.2 x 10e-3 N*m*s wheel momentum. This would create a slew rate at around 17 minutes with a 0.2 degrees steady state error.

Acknowledgements

I would like to thank my advisor, Dr. Periklis Papadopoulos, for the guidance and support over the course of this project. I would also like to thank my family and peers for the encouragement that help me continue my ambition for aerospace.

Table of Contents

List of Figures

List of Tables

Symbols

Symbol	Definition	Units (SI)
A	Reference (projected) surface area exposed to a flow or radiation	m^2
B_{Earth}	Local geomagnetic-field vector	T(tesla)
c	Speed of light in vacuum	m/s
C_d	Aerodynamic drag coefficient	—
C_{pa}	Centre-of-pressure position vector	m
C_m	Centre-of-mass position vector	m
F_s	Solar-radiation force magnitude on a surface	N
H	Angular momentum of spacecraft + wheels	kg m^2 s^{-1}
h_w	Wheel-only angular momentum	kg m^2 s^{-1}
I	Inertia tensor of whole spacecraft in body frame	kg m^2
I_w	Inertia tensor of a single reaction wheel	kg m^2
I_{SRP}	Solar-flux density	W m^{-2}
K	Boltzmann constant	J K^{-1}
M_{sat}	Satellite residual magnetic dipole moment	A m^2
m_{north}	Unit vector toward magnetic-north pole	—
q	Attitude quaternion	—
r	Spacecraft's position vector from Earth center	m
R_{earth}	Mean Earth radius	m
T_0	Standard noise-temperature	K
V	Spacecraft orbital velocity magnitude	m s^{-1}
ω	Angular-velocity vector of spacecraft body	rad s^{-1}
ω_{rw}	Spin speed of a reaction wheel	rad s^{-1}
τ	External or control torque vector	N m
Greek Symbols		
α	Sun-incidence angle on a surface	rad
β	Azimuth angle of individual wheel in the array	rad
γ	Azimuth angle of individual wheel in array (0°, 90°, 180°, 270°)	-------
μ_{Earth}	Ratio of reflected to incident solar energy	m^3 s^{-2}
ρ	Earth gravitational parameter (GM)	kg m^{-3}
σ_n	Atmospheric mass density in LEO	rad s^{-1} $\sqrt{Hz^{-1}}$
τ_{SRP}	Gyro bias-instability standard deviation	N m
τ_g	Solar-radiation-pressure torque	N m
τ_{res}	Gravity-gradient torque	N m
Acronyms		
ADCS	Attitude Determination and Control System	-------
HWIL	Hardware in the Loop	-------
IMU	Inertial Measurement Unit	-------
MCU	Microcontroller Unit	-------
PCB	Printed Circuit Board	-------
LEO	Low Eart Orbit	-------
I2C	Integrated Circuit	-------
MEMS	Microelectromechanical systems	-------
MEKF	Multiplicative Extended Kalman Filter	-------
ECI	Earth-centered inertial	-------
ECEF	Earth-centered Earth-fixed	-------

1. Introduction

1.1 Motivation

The current state of the art of the altitude determination control systems composed of actuators such reaction wheels, sensors such as gyroscopes, accelerometers, star tracker and sun sensor. These systems combined would allow extreme fine control for optical and communications pointing application. As satellite orbits Earth, there are several disturbances torques that are applied within its atmospheric, magnetic and gravitational conditions. A well-qualified ADCS systems is capable of stabilizing and detumbling the spacecraft according to its mission profile. For large spacecraft bus in the geosynchronous orbit applications these elements tend to be distributed from several spaces In the spacecraft since the size of the components are too big.

However, smaller CubeSat application under 12U needs elements to be more compacted and extremely reliable for the environment of space. The same approach of building and tailoring CubeSat's ADCS systems has been proven to be too expensive for its desired profile which make the mission more expensive. The power requirements of these extremely reliable approach have required too much power consumption which lead to high power systems budget for the attitude determination and control system (ADCS) especially the actors and the inertial measurement unit (IMU).

Several Attitude Determination and Control System (ADCS) configurations are currently being explored to optimize the control feedback mechanisms and reduce the overall mass of the subsystems. Configurations under consideration include tetrahedral, pyramidal, and skew arrangements of reaction wheels. Each of these designs is evaluated for its potential to enhance stability, control precision, and minimize the mass properties, which are critical factors in small satellite missions. There have been consideration to use high grade consumer electronics components to also cut down the cost [1].

To further compact the system, sun sensors could be integrated in near proximity to the actuators. This integration not only streamlines the ADCS architecture but also enhances the accuracy of attitude determination by reducing the time delay between sensor measurements and actuator response.

Previous design of 0.5U compact systems need to go through a more through trade study as well as optimization for reliability. This approach necessitates that the CubeSat's chassis be designed around the ADCS, with particular attention to the coupling between the control system and the optical or instrument payload that the CubeSat carries. By doing so, the structural design would inherently support the dynamic requirements of the ADCS while also ensuring that the payload remains stable and properly aligned during the mission. This holistic design strategy would potentially lead to a more efficient, compact, and lightweight CubeSat, maximizing both its functionality and performance in space.

1.2 Literature Review

The overview of the literature review will be involved in a big picture subsystem of subsystems architecture decomposition of the Attitude Determination Control Systems. This includes the aspect of attitude determination of sun sensor, inertial measurement unit, and magnetometer. For attitude control aspect will cover reaction wheel/motor, hardware in the loop integration/testing, and the electronic reliability research have been performed on cost off the shelfs components. This literature review will attempt to cover the current subsystem component level challenges and identified the possible roadmaps as well as reliable existing components/systems. There will also be past critical design technical review report to be analyzed for operational feasibility.

1.2.1 Attitude Determination Aspect

The scheme of attitude determination are defined by multiple onboard sensors that make up the process of navigating the location and states of the vehicle during operations.

1.2.1.1 Digital/Analog Sun Sensors

Sun sensors are implemented to estimate and provide feedback of the direction of the Sun with respect to the spacecraft's body frame for attitude determination. Modules of sun sensors are used for mission in LEO CubeSat instead of star tracker since they tend to be less expensive and more reliable. Sun sensors are detectors of infrared lights or within any other visible light spectrum with the function of measuring more than one to two sets of mounting bases and the incident incoming sunlight. The architecture of the sun sensor essentially uses photodiodes to detect light rays or photons from the sun where it the convert to digital or analog based signal for intensity presentation. The challenges of current design of sun sensor for CubeSat to meet the less than 5 degrees requirement is that they are very expensive to manufacture. So, one of the current techniques to make it cheaper to make an array of photodiodes with multiple cells to make it cheap despite sacrificing the mass [2].

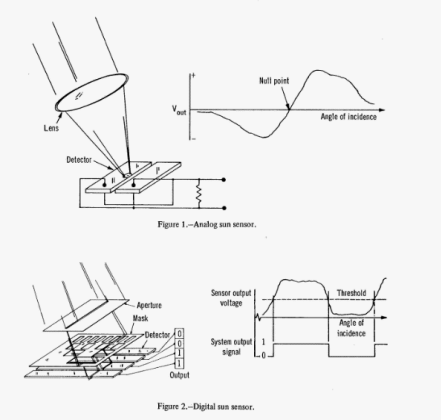


Figure 1.1. Analog vs digital Sun sensor [3].

There are two main types of miniature sun sensors configuration such are cosine detectors and quadrant detectors. To solve the compact issue and achieve low-cost quadrant detectors are much preferred in the current state of art. The main difference of quadrant detectors and cosine detectors is that quadrant sensors operate through a 2x2 array of photodiodes. While the cosine photocells use the current generated by the cell which is proportional to the cosine of the angle of incident of Sun rays. This means that they would need 4 suns sensors to obtain accurate coverage. As for specifications of CubeSat payload the desired sun sensor would have an accuracy of less than 5 degrees.

As for the modeling and the algorithm computation of the sensor, the only challenge is the placement of the sensor such that it will follow the design of the array of photodiodes [4]. The incident angle can be calculated by processing the output detection or using a co-planar array. Filter can also be added to improve the characteristic of the Field of view and transfer function as well as reduce the electromagnetic interference by adhering to a good electronics packaging standard [5].

1.2.1.2 Inertial Measurement Unit

The current state of art of IMU measures and reports on the spacecraft's positions and acceleration by using the gyroscope and accelerometers micro-electromechanical systems (MEMS). Their readout rate is driven by the parameters of acceleration, angular velocity and magnetic field strength if magnetometer is involved to create a 9 Axis unit. It is part of the critical control systems feedback to maintain or change the orientation for operation commands. The current challenge for IMU in CubeSat especially for 9-Axis module is the extreme size of miniaturization and integration. Such problem for 9 axes come into the minimal power drawn and still be able to achieve accurate data and cost of precision. During integration process, the modularity of IMU is extremely important since they must be able to do unit test then immediately plug into the systems for the systems level test. Another challenge during operation launches the IMU experience shock and significant vibration therefore IMU packaging need to be robust to manage this force without constant calibration [6].

Current state of the art of IMU addressing the noise problem is to implement a technique called sensor fusion which means carrying multiple redundant IMUs as well as magnetometer on a different board to reduce coupling interference. Such filter also must be implemented in the algorithm to be surely and well calibrated at the unit test level. The MEMS packaging technique involved in semiconductor manufacturing process reduce the power consumptions and include more transistors than ever before which make bias stability metric extremely reliable [7].

1.2.1.3 Magnetometer

Magnetometer is an essential element of the CubeSat's orientation determination respect to the Earth's magnetic field where the formation precision is critical for operation such as instruments and antennas telecommunications. Many missions use magnetic field mapping to prepare for the mission profile and trajectory. Magnetometer is critically used in sensor fusion to eliminate the error of drifting if using IMU as a sense of redundancy. One of the biggest challenges is the magnetic shielding interference from other on-board electronics such concerns are isolation which tend to reduce the accuracy of magnetometer which desensitize the accuracy of the sensor fusion process [8].

This challenge can be addressed by using onboard magnetic packaging shielding such that the clipping architecture will cover the entire integrated circuits to prevent leakage of magnetic signal at MHz operation. There can also be a time-varying bias be corrected by compensation methods during hardware I the loop testing. There are also techniques used called ferromagnetic materials to account for distortion known as soft and hard iron effects to help compensate the readings. This therefore would couple with the location of the arranged in the compact ADCS systems where magnetometer would be placed as far from the mechanism or fast switching mechanism as possible. Such location can be found in the middle of the printed circuit board [9].

1.2.1.4 Attitude Control Aspect

Attitude determination and navigation allow the spacecraft the objectives within its concept of operation to feedback to the controller maneuvering operation of the spacecraft to the desired directions. This is where attitude control aspect comes into play with multiple methods for different applications. Many modern control techniques require fine or coarse pointing due to the type of payloads the spacecraft is carrying.

1.2.1.5 Reaction Wheels and Motors

The reaction wheels subsystems are devices that provide active attitude control and serve as stabilization torque on the spacecraft. Depending on the moments of inertia from the design, they will have a peak momentum and maximum torque. By adding current to the motor which alternate the revolution per minutes allowing different axis to rotate assuming the reaction wheel fully align on the axis center coordinate frame [10]. The wheels can spin at different speed by digital pulse wave modulation form provide by the brushless direct current motor. This is one of the biggest challenges for CubeSat since the manufacturing tolerance and the redundancy won't be as high due to cost saving compared to advanced big bus spacecraft. Therefore, having only 3

reaction wheels for each axis is a risk to be mitigated. Fortunately, the cheapest and most reliable solution that the industry is heading toward is to implement the four reaction wheels configuration for fault tolerance. Such configuration can be described as tetrahedral or skewed configuration in Figure 1.2.

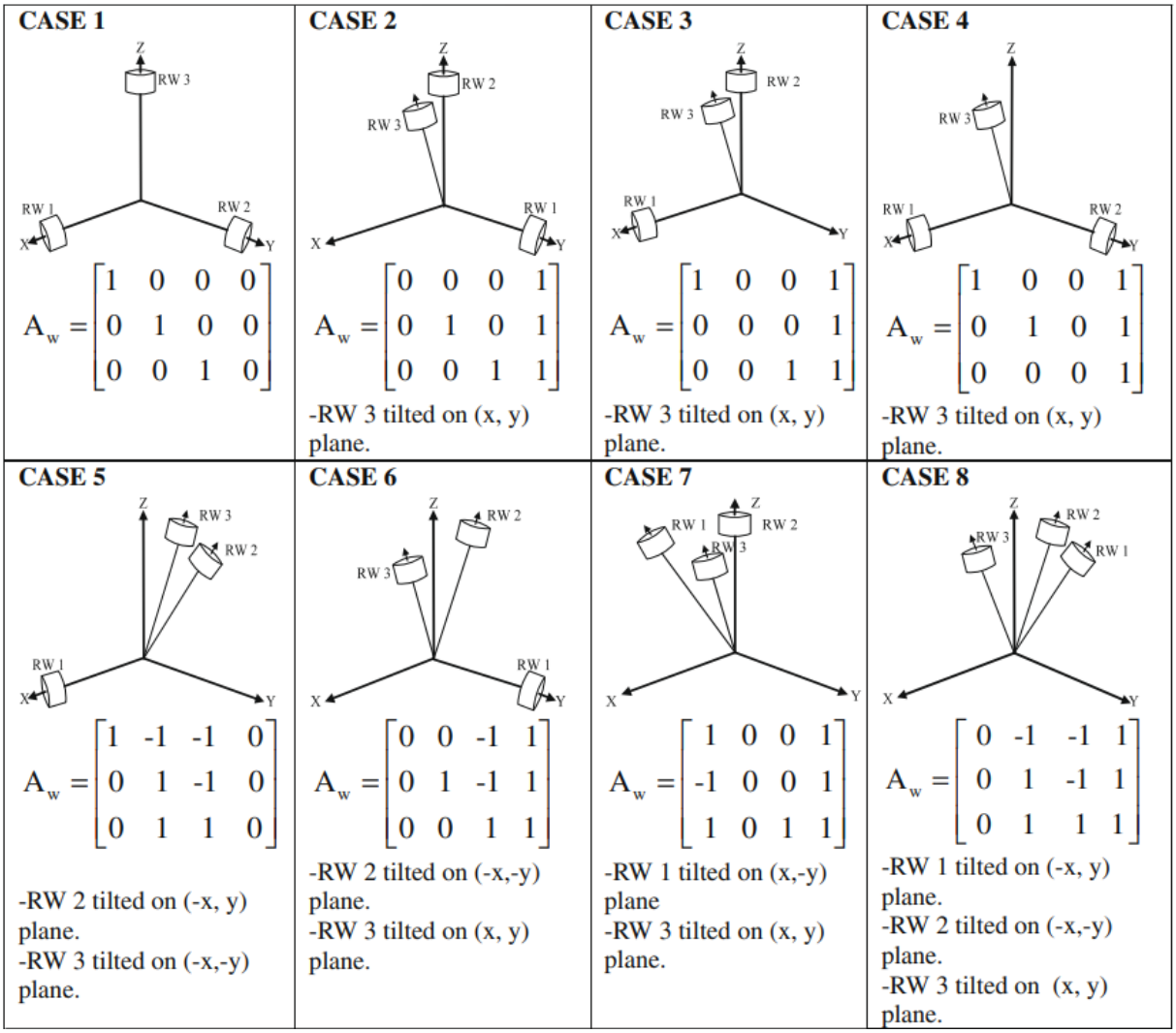


Figure 1.2. Four reaction wheels configuration [11].

These arrangements of not aligning on the principal axes of the CubeSat main serve to create cross-coupling of x, y, and z components of the control dynamics scheme. The tradeoffs are that more power would need to be drawn to maintain same torque desired since the reaction wheels would all need to work together to perform spins [12]. This also reduce the torque on a single access as it allows redundancy for all three axes. Many of current state of the art use extremely tight electronics packages such as Blue Canyon Technologies reaction wheel where their mass is quite heavy however producing good peak torque and high momentum capacity [13].

Another challenge for motor miniaturization and compact integration is that the vibrations of four reactions wheel create significant interference when different modes of

operations are being switched. This occur when microcontroller send different signal to four motors with digital waveform producing electromagnetic interference to multiple sensors on board. This therefore require an Emi coating for isolation. For motors reliability the most concerning challenge is the wear and tear of the mechanical bearings with friction. Therefore, even though it is preferrable to used Brushless DC Motor there still need to be space lubricants for vacuum environment to prevent outgas that might condense on the payload which will jeopardize the mission. Another reason why Brushless DC motor is preferred is that it uses a three stages driving coils along with hall sensor or optical encoder to determine the rotor position when activating the coil phases [14].

Mounting the wheel on the miniaturized motor's shaft for CubeSat manufacturing and integration process is the biggest mechanical challenges due to its coupling to vibration effects. The alignment and balance must be perfectly balanced during rotation and the securing of the wheel must not offset and move along the shaft's axis which will change the moment of inertia of the systems. To solve the issue of integration there has been several proven methods of using press fit and interference fit to fully secure the wheel. This provides accurate and solid connection without additional fasteners which decrease the mass. Another method is to use a set screw or gur hub to hold it in place however that will not be able to use for high torque application. Therefore, interference fit is the most viable option as well as laser welding to enhance the rigidity of the assembly [15].

1.2.1.6 Hardware-in-the-loop and Control

Hardware-in-the-loop simulation testing and control is a crucial part to verify/validate the embedded systems within the desired controlled environment. For any ADCS systems the testing involves at the systems level to seek responses under realistic conditions before deployment. The current development of cheap hardware in the loop (HIL) topology is a challenge since the expenses of equipment are quite high to capture everything within the real time operating systems. Thus, modeling a high-fidelity test stands and imitating the exact conditions of orbit is immensely challenging for small CubeSat project. The most notable subsystems that is challenging to perform HIL testing is the magnetorquer as well as reaction wheel since they required Helmholtz cage as well as air bearing fixtures which are very expensive to manufacture. A cheap current state of the art of a reaction wheel test systems is to introduce a gravity free systems and non-friction system through an air compressor. That air compressor must be filtered by an air dryer, air filter and air particle filter to reduce the moisture and fine particle which might turn int external forces or disturbances interfering with the results. The air in which are being blown into the hemispherical air bearing below the satellite ADCS mounted platform in Figure 1.3 [16].

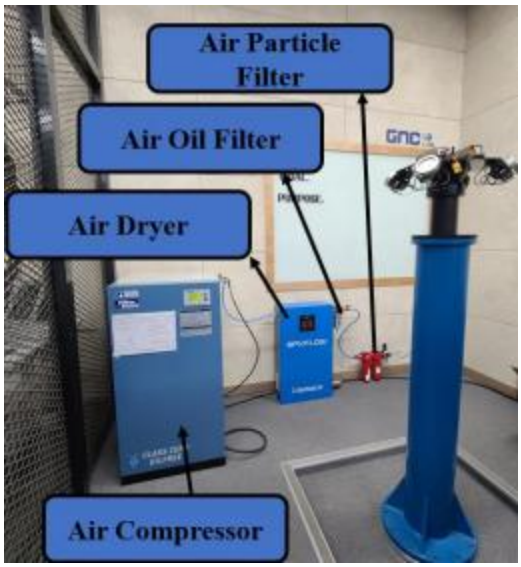


Figure 1.3. Reaction wheel HIL systems [16].

As of the control of the hardware, the commands are generated by the main board microcontroller which defines a set of torque with the respective quaternion inputs of desired attitude into the controller where the plants are the reaction wheels with introduction of disturbances torques that loop to the IMU sensors as well as determination algorithm such as Kalman filters. There can also be sensor fusion as part of the hardware control loop. The rotation of the module can be measured through standstill location. While the magnetometer and sun sensor can be integrated with each other. Triad technique to form a vector to verify the current position. After an intermediate position is determined, the magnetic vector and sun vector can be computed and passed to Kalman filter to resolve an accurate state estimation in noisy conditions. Often there will be a set of numerical simulations that feed to the simulated models the need into the hardware to complete the loop. However, such models requires very accurate depiction of the state as well as fine characteristics. The visualization and collection of telemetry will be displayed on application such as Grafana or Telemetry viewer since they are able to be accessed by serial ports communications [17].

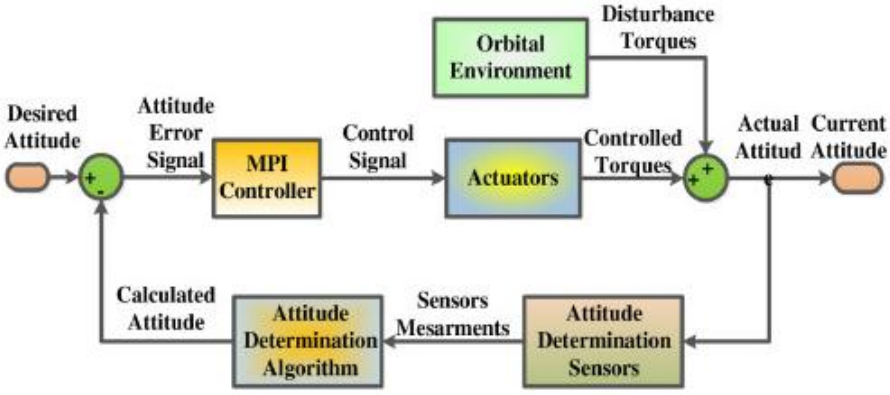


Figure 1.4. Topology of hardware feedback loop [17].

1.2.2 Electronics Packaging

Despite Miniaturization and compact design of space hardware saved mass and space usage, the effects is that it leads to increase in power density which makes thermal management become much more challenging. Due to limited surface area for heat dissipation passive cooling is essentially less effective and need the inference of active cooling. There is also a concern of thermal expansion which jeopardized the integrated circuit structural integrity over a wide temperature range from -40 degrees Celsius to 125 degrees Celsius. The most effective cooling methods that doesn't add mass is to use high performing thermal interface material to distribute heats more evenly and will also focus on the designer of the printed circuit board that host these components. The junctions of the integrate circuit interface with the circuit board or to a heatsink that use radiative cooling which maximize the surface area exposed to space environment to radiate heat away from the electronic components [18].

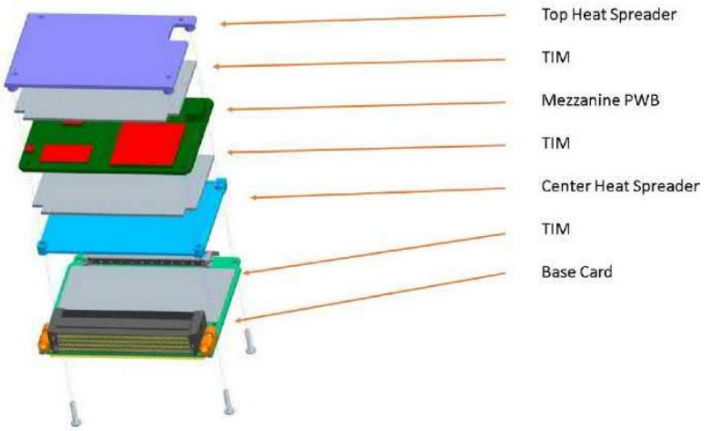


Figure 1.5. Topology of avionics card stack [18].

Such examples of usage of thermal interface material and heat spreader can be depicted in Figure 1.2.3. The TIM material in this cased are used to fill the microscopic air gaps between surfaces improving thermal conductivity between the card and the heat spreader.

Compact miniaturization concept also posed a vibration during launch which might create creeps or voids of the solder package. This is essentially display as a risk toward the photodiode sun sensor board. As the electronics are compact, the power density increases which couple into fatigue which also lead to cracks within the solder joints. One of the key methods is to use a dampening material as well as determine the load paths for PCB mounting. This would go hands in hands with the current industry proposed fasteners selections. Many of past research have leveraged FEA to predict the resonances to iterate their design much faster. The common suggested architecture has been to not place heavy components near the edge of PCB which can amplify the vibrations as well as to spread the weights of components to be even as much as possible. Achieving proper torque and preload will also be able to secure the PCB on the electronics chassis. Certain tests that need to be simulated are random vibrations, sinusoidal and shock vibrations according to the GEVS standard [18].

1.3 Project Proposal

The proposed project is to complete and optimize the development and integration of the custom compact attitude determination control systems. The project objective is to bring proof of concept of low cast and highly reliable electro0mechanical electronics architecture into the reaction wheel systems. There will also be a sun sensor board and magnetometer electronics board to complete the fundamentals of the ADCS control feedback loop systems. There will be three phrases to the projects which are hardware physical and model-based development, hardware in the loop control implementation, and verification & validation.

1.3.1 Hardware and Model-Based Development

Model based architecture of the design is going to be captured in both Activity and Sequencing diagrams. The goal is to conduct a detailed analysis and thorough design parameters to come up with proper specifications require which is angular momentum and torque output with reference of minimizing power input. The modeling of sensors such as internal measurement unit, sun sensor, and temperature sensors are very critical for sensor fusions methodology control.

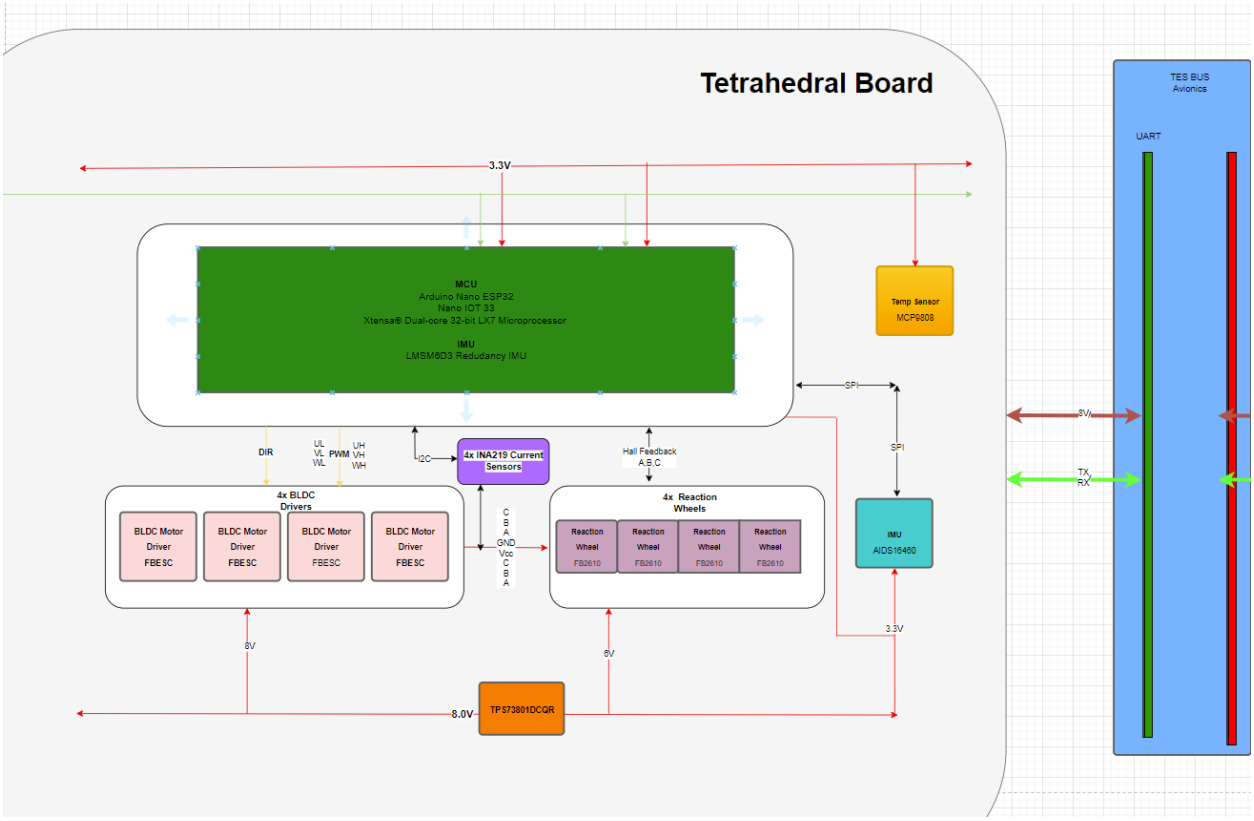


Figure 1.6. Electrical topology of the tetrahedral board.

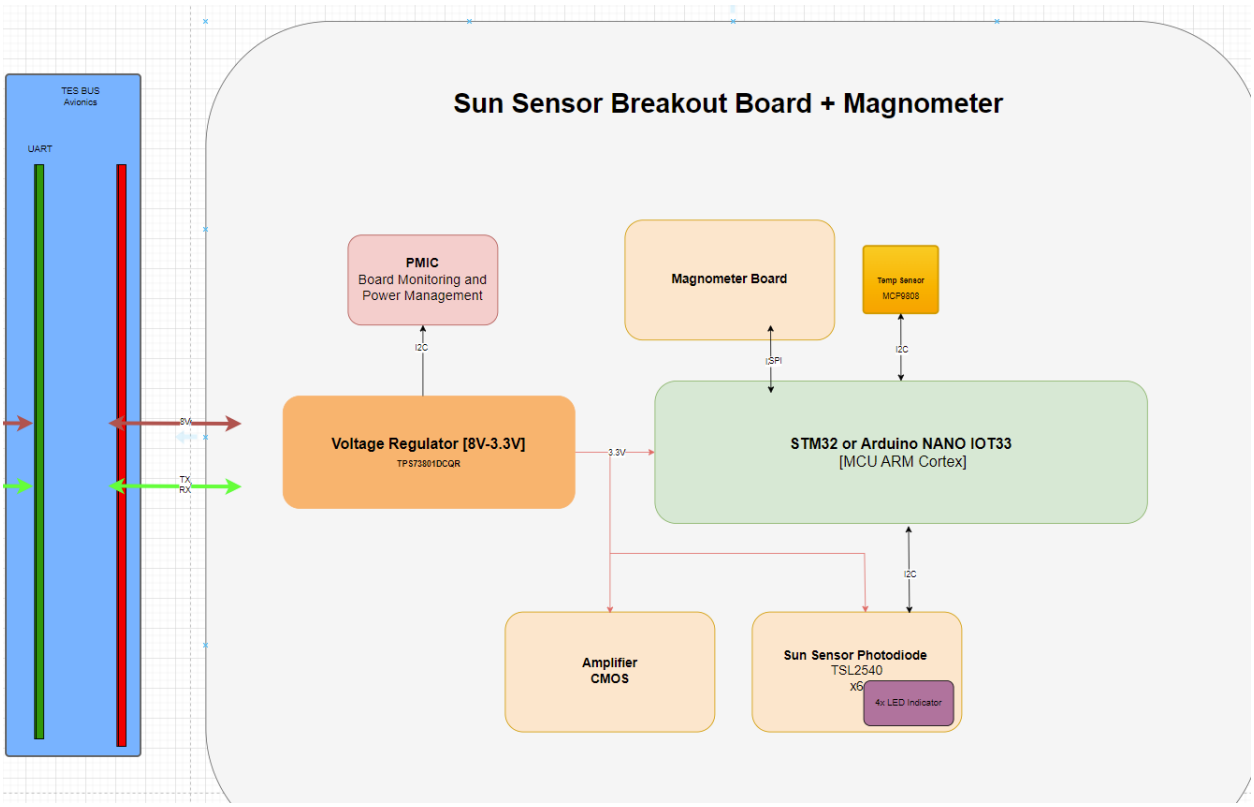


Figure 1.7. Electrical topology of sun sensor host board.

As shown in Figure 1.2. Electrical Topology of Tetrahedral Board and Figure 1.3. Electrical Topology of Sun Sensor Host Board; both boards are connected to a UART and a Power rail to talk with the main CubeSat bus. There will be a model to dive deep into the definition of each component and their interfaces. The design of the physical hardware will follow the tetrahedral configuration mounting for controls redundancy as shown in Figure 1.2.1.

The sun sensor board will compose of photodiodes quadrants using digital process output of intensity. Tetrahedral reaction will board will provide the control as well as the IMU host to feedback into the brushless direct current motor.

1.3.2 Hardware -in-the-loop (HIL) and Control Implementation

The objective is to develop and control algorithms for reaction wheel systems with the emphasis of PID control schemes. The inputs and model will be based on quaternions for precision control. Sensor fusion and Kalman Filter implementation is also one of the goals to enhance the accuracy of the feedback data.

1.3.3 Verification and Validation

The performance evaluation will be based on the angular and torque output as the specification of the project. There is also other specifications such as mass, power consumption and reliability as an acceptance criteria for the production. Thermal reliability of the electronic

components and architecture are evaluated under multiple printed assembly configuration. The vibration analysis will need to be performed as a qualification criterion for the product across with simulation analysis.

1.4 Methodology

1.4.1 Model Based Hardware Development

The Model Based systems architecture will be done on Cameo Systems Modeler. The block definition diagram are composed of major and minor components with their stereotypes and properties. Such attributions to the architectures are accuracy, range, power consumptions, mass, and thermal range. The second component will also be established to display the association of each block dependencies as well as the high-level flow down. The internal/interface block diagrams will contain the ports and systems activity that contain the data packages for each of the operation.

The electrical components such as the reaction wheel central printed circuit board and sun sensor circuit board host will be created in and Electronic Design Automation software called KiCad and will be fabricated through PCBWay. For the mechanical mounts and structures will be 3D printed with thermoplastics such as Nylon for prototype purposes. Fasteners and joints will be traded on threaded inserts assembly.

1.4.2 Hardware-in-the-loop Control Implementation

The hardware in the loop scheme will be implemented in Matlab and Python. There are telemetry Viewer and Grafana applications as visualization of the telemetry output connected to the main hardware. There will be controls libraries leveraged to input quaternions model for hardware parameters evaluation both units test and systems test.

1.4.3 Verification & Validation

For thermal reliability, a full model will be model and simulated in SolidWorks to simulate burn in conditions of the integrated circuit on the circuit board. Ansys or SolidWorks will be used to simulate the random vibration, shock and possible sine sweep operational conditions. Hand Calculations will also be used to verify and validate the results convergence [19].

2. Attitude Determination and Control Subsystems Design

2.1 Electrical Hardware

The electrical hardware architecture of the Ventura ADCS systems are composed of two independent printed circuit board assemblies which are the reaction wheel control wheel board and sun sensing determination board.

2.1.1 Microcontroller

The microcontroller unit is a motorized integrated circuit to command, control and handle operations in an embedded system. A lower-level decompositions of a microcontroller includes a processor, input/output[I/O], peripherals and memory on a single chip. It is like a computer where it can connect to multiple sensors and collect data for a certain desired application. In this usage, the microcontroller is used to capture telemetry feedback for the feedback control loops.

Conducting trade studies involving understanding several parameters include:

- Mass: The physical weight derived from the dimension of microcontroller which is sensitive for CubeSat applications
- Power Consumption: The amount of power required which affect systems efficiency and on-board battery
- Operating frequency: The speed at which microcontroller unit execute and handle instructions which typically measured in MHz or GHz which also impact power drawn.
- Memory Size: The amount of available of RAM and Flash memory for the uploaded embedded firmware code
- Independent Watchdogs: The mechanism that help radiation tolerancing
- Debugging features: The feature of JRAG that help with troubleshooting and debugging
- Interfaces drivers: I2C, SPI and UART application to communicate with another peripheral.
- Clock speed: The rate in which how quickly the controller can process instructions.

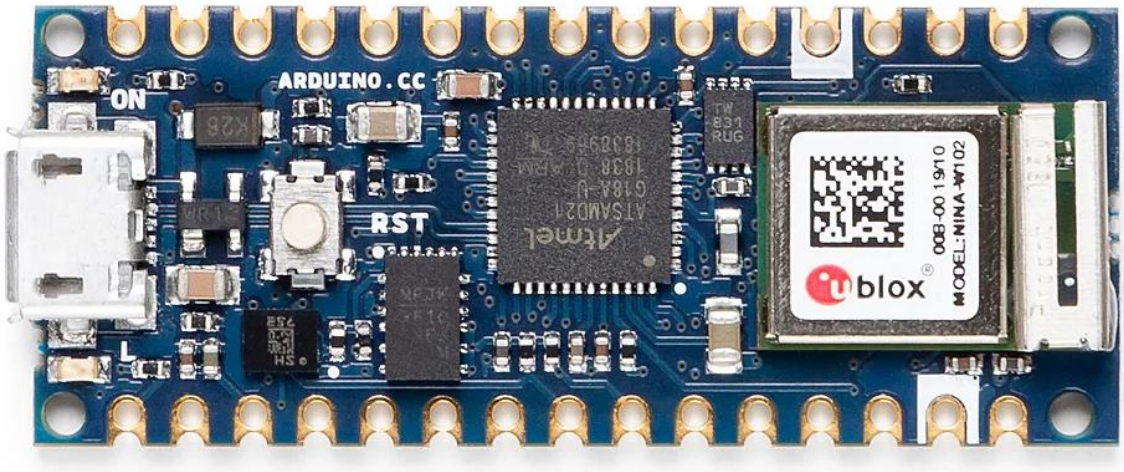


Figure 2.1. Arduino Nano IOT33 [20].

Considering the trade parameters, the table below has been created for component down selection.

Table 2.1. Specifications for the Arduino Nano Iot33 and STM32F103C8T6 (Blue Pill) microcontrollers.

Parameters	**Arduino Nano Iot33**	**Blue pill**
Mass	7 grams	20 grams
Operating Frequency	48Mhz	32Mhz
Flash Memory	256kB	64kB
Operating Voltage	3.3V-5.0V	3.3V-5.0V
Analog Pins	8	10
Digital I/O Pins	14	29
I2C Buses	1	2
SPI Buses	1	2
Form Factor	18x45mm	23x53mm

From the table, it is clear to see that the Arduino Nano offer much better mass and form factor for the miniaturization purposes. Despite, the Arduino nano offer fewer pins compared to the blue pills, the Ventura ADCS control systems utilize the exact 14 pins on the platform which is extremely efficient for the bs communications. It is important to note that it is required to operate on both I2C and SPI communications protocol since high speed and sensitive component such as the inertial measurement unit needed a good isolation line commonly preference would be SPI protocol.

2.1.2 Inertial Measurement Unit

The IMU is a micro-electromechanical systems that obtain the telemetry and attitude data foe the ADCs. The subcomponents include accelerometers, gyroscope and sometimes magnetometer. Accelerometers monitor the velocity and acceleration of the body. Gyroscope

determine the rotation and rotational angular velocity rate. Magnetometers find the cardinal direction which sensing the Earth Magnetic field. These devices would often be used together to correct fot bias drift during operations. IMU's weakness during operation is that it often accumulates error which create dtft since I<U constantly monitor changes relative to itself, so it consistently computes rounding errors within its computation which accumulate over time. A common method algorithm is sensor fusion technique to mitigate the offset problem.

On the Ventura ADCS board, there are two IMUs. ADIS16460 which are from Analog Devices is being used as a primary IMU. LSM6DS3 from TDKVsense is the secondary redundancy IMU in case the primary IMU malfunction.

Table 2.2. Specifications for the ADIS16460 and LSM6DS3 IMUs.

Parameters	**ADIS16460**	LSM6DS3
ARW(Angular Random Walk)	0.015-0.023 deg/sec/Hz	0.015-0.023 deg/sec/Hz
Bias Instability	16 deg/hour	5 deg/hour
Vibration Rejection	0.01/sec/g	8g
Temperature Range	-25C – 85C	-40C-85C
Operating Voltage	3.15V	3.3V

Figure 2.2 ADIS16460 IMU [21].

The ADIS16460 was chosen to be the primary IMU since it offers very high ARW and bias instability which makes it very resilient for the mission profile up to 7 months. The way the electronics packing of MEMS of the ADIS16460 is extremely more ruggedized to operate in a harsh environmental condition. LSM IMU will be used as backup and operate at much lower voltage. This is to help AIS16460 with the cross-reference data readings before feeding into the control loop. Both IMUs, are leveraged with sensor fusion techniques to process sensor data to the microprocess MCU for quaternions formation as well as Euler vector rates in a very fast and accurate manner.

It is critical to quantify the bias error noises and source such that it is a non-zero output for a zero input rate and an offset in output rate from the input rate which has the unit of (degree/hour). The bias instability parameter o the other hand are random variation in bias or offset over time which has the unit of (degree/hour). The angular random walk represents the drift rate and angular error due to White noise I the angular rare which has the unit of (degree/$hour^{1/2}$). The scale factor error is another parameter that shows a deterministic error in the measured angle of rotation linearly proportional to the angle of rotation (ppm). The difference in the measured angle of rotation in the CW and CCW directions are linearly proportional to the angle of rotation.

2.1.3 Magnetometer

Magnetometer on the Ventura ADCS systems is selected and required to provide detection and sensitivity on XY and Z axis. They are controlled and riven by the MagI2C controller connected to the coil sensors provided by Pni manufacturer.

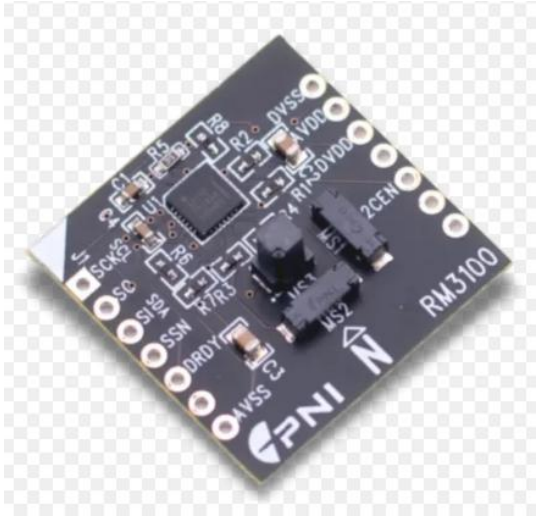


Figure 2.3. RM3100 breakout board [22].

Table 2.3. RM3100 specifications.

Sensitivity	13 nT
Noise	15 nT
Sensitivity Sampling Rate	300 Hz
Voltage	3.3 V
Operating Temperature	-40°C to 85°C

The main reason why RM3100 is needed is for the sensor fusion heading direction of the ADCS algorithm. It offers very high precision magnetic field measurement which high sensitivity where small magnetic field fluctuations are hard to detected in this case is Earth. The current challenge Ventura is facing is the electronics packaging for the magnetic field isolation away from components such as IMU and high switching devices. Therefore, the RM3100 will not be hosted on the Ventura PCB board, instead it will be on an isolated EMI shielding

compacted mounted feature on the 6U TechEd Sat bus where there is minimal electronics presence.

The magnetic reference algorithms provide coarse spacecraft attitude knowledge via magnetometer with the algorithm compares the measured magnetic field with a predicted B-field from the on-board Earth magnetic field model. There is an instantaneous 3-axis attitude knowledge that can be derived from this B-field comparison. Magnetometer measurements are used for magnetic momentum management, typically external torque source actuating of the middle. The torque rods use do to dissipate excess systems momentum by impairing tan external torque of the spacecraft.

2.1.4 Sun Sensor

The selected sun sensor to be used is BH1730FVC. BH1730FVC is a digital ambient light sensor that has a very wide detection range for the sun position vector. It offers very low power consumptions. Arranging this in an array platform provide the incident of sunlight with respect to the spacecraft's body. This also increase the field of view wider angular range. The goal is to simulate the 4 quadrants just like the ADCS reaction wheel tetrahedral mounting configuration. This designed is proved to cover the full hemisphere allowing the spacecraft to detect the sun intensity even when the spacecraft is performing its operation at an oblique angle.

Since this is a consumer electronics grade components there need to be reliability and redundancy. This is where the arrays come in, where if one photodiode fails, the rest of the array can still function across the quadrant of the spacecraft. The design to hope to also represent the intensity by varying the LEDs on the printed circuit board as a form of verification/validation method.

Figure 2.4. BH1730FVC sun sensor [23].

Table 2.4. BH1730FVC specifications.

Sensitivity	13 nT
Detection Range	0.001 Ix to 100k Ix
Sensitivity Sampling Rate	300 Hz
Voltage	3.3 V
Operating Temperature	-40°C to 85°C

This is simply a photovoltaic cell that produces an output current when illuminated by sunlight. The current is proportional to cosine of the sun angle (projection of area to sun is the cosine function like in Eq. (2.1). The cell produces no polarity or directional information. All directions with same half-cone angle produce same output current.

$$I = I_o * \cos(\phi) \tag{2.1}$$

Where ϕ is the sun angle of incidence, and I_o is the max cell output current.

2.1.5 Temperature Sensor

The down selected temperature sensors that exist on bot Ventura ADCS and Sun Sensor board is MCP9808. The MCP9808 has been used in various CubeSat mission such as AcubeSAT in Greece. The main reason is the very low power consumption which allow active power shutoff mode. Placing this very versatile in conjunction with the chassis and on PCB allow for traceability of heat sources/sink which enhance the process of anomaly detection. The usage of MCP is also used to help compensate for Sensor drift in case the IMU experience drift due to their temperature changes. The placement of the temperature sensor need to be strategically placed near heat sensitive component where it is sensitive to heat fluctuation.

Figure 2.5. MCP98089 temperature sensor board [24].

Table 2.5. MCP9808 specifications.

Sensing Range	-40C – 125C
Precision	0.0625 Degrees
Voltage Operation	3.3V
Current	200uA
Form Factor	21mm xx 13mm x 2mm

2.1.6 Electromechanical Motor

There are four motors mounted on the tetrahedral configurations of the ADCS module. Each are capable of controlling two axes for redundancy purposes. This means that there are a certain distribution of torque to rotate the entire CubeSat as well as to respond to the external disturbances.

There are two types of motors in the cost of the shelfs market which are Brushed DC motors and Brushless DC motor. The Brushed DC motor contain mechanical commutation through the brushes and a commutator that acts as a two-pole magnet. As the motor operates, the brushes conduct current to the armature windings via the commutator creating the magnetic field producing torque along the stator's magnet. The directions of the motor can be changed due to its polarity of the electromagnet in the armature. On the other hand, the brushless DC motor does not use mechanical commutator by using a speed controller. It contains three stages of coil as a driver as well as containing hall sensor to measure the exact rotor position.

There are many advantages and disadvantages that can be conducted via the following trade study ranking matrix:

Table 2.6. BLDC trade card specifications.

Parameter	Brushed DC	Score	BLDC	Score	Justification
Efficiency	Lower [70%]	2	High [90%]	5	Brushless are significantly more efficient due to less friction and better energy conservation
Torque	Medium-High Initiating Torque	5	Consistent Torque	4	
Speed Control	Simple Phase	3	Precise Control	5	Brushless DC offer finer precision control over wide speed range

Lifetime Reliability	Lower	2	Higher with less mechanical part wear/tear	5	Brushless DC is more reliable since it doesn't contain many wear/tears mechanism
Cost	Lower initial cost	5	Higher initial cost with internal electronics	2	Brushed motor is more affordable since brushless contain more advanced electronics
EMI	Higher emission	2	Minimal emission	5	Brushless motors produce less EMI due to no commutative coil
Thermal	More hear generated	2	Better thermal performance	4	Brushless motor generates less heat since they have higher efficiency
Power Consumption	Power consumptions	2	Lower consumption	5	Brushless conserve energy which draws more current
Total Score		23		35	

The down selected motor for the Ventura ADCS module is Faulhaber 2610C [20]. This is a brushless DC motor that has an integrated speed controller within the housing to reduce the printed circuit board footprint. The aim of the torque budget is to have at least one mNm per axis contribution of rotation. Therefore, the essential parameter to focus on is the output and stall torque. The following table portrays the characteristic of the motor:

Figure 2.6. BLDC FB2610 Motor [24].

Table 2.6. FB2610 specifications.

Voltage Range	4-18 V DC
Nominal Voltage	6V
No Load Speed	6700 – 13,000 RPM
No Load Current	0.02A
Maximum Efficiency	95%
Maximum Efficiency Speed	400-13300 RPM
Stall Torque	17.6 mNm
Stall Current	0.53 A
Mass	20.1 g

The concept of back EMF is the voltage generated across the motor; s terminals as the armature moves though the motor's magnetic field. The back EMF opposes the drive voltage and is proportional to the motor speed velocity such that:

$$V_{EMF} = K_{EMF} * \omega \quad (2.2)$$

As the wheel speed increases the V_{EMF} increases decreasing the voltage across the motor with the following impacts if there is not enough bus voltage overhead. Decreasing the voltage across the moto decrease the torque until torque =0. The wheel can no longer accelerate limiting maximum wheel speed.

2.2 Connectors and Harnesses

2.2.1 Connectors

Connectors are the crucial interface points or nodes for every electrical power and signals between the Ventura or Sun Sensor Board to their respective peripherals as well as the main on board computer. These components need to be able to withstand all sort of mechanical shock, vibration and thermal load for mission reliability.

Table 2.7. Connectors and applications on the circuit board.

Specification	Molex Pico-Lock Connector	Molex Picoblade 1.25mm Connector	SMT Right-Angle Tin-Plated Connector	Phoenix Terminal Block Connector
Pitch[mm]	1.5	1.25	1.5	COMBUCON
Current Rating [A]	3A	1A	2A	15A
Number of Position	6	6	6	5

Operating Temperature (Celsius)	-40 C to 105C	-40C to 85C	-55 C to 125C	-40C to 105C
Mounting Type	SMT	SMT	SMT	Through Hole
Usage Application	Power/Signal Motor	Power/Signal Motor + Magnetometer	Power/Signal	Power/UART

The Pico-Lock Connectors were selected for their compact form factor and reliable electrical power and signal performance. These connectors offer locking mechanism in a low-profile design which has proven to survive launch and low earth orbit environment. The secure mating allow the risk of mechanical shock as well as very integrated friendly by surface mount technology for rapid production. The Molex PicoBlade connector of 1.25 mm on the other hand is mainly for the electrical signal connections that require smaller pitch and lower current rating. This allow the versatile of different configurations and enhanced quickness in early development phase. For Surface Mount Right Angle Connectors, this is utilizing the temporary through hole as well as tin plating for sufficient corrosion resistance for the intended operational environment

The Phoenix Terminal Block connector is integrated for the stability of power connection that require higher current handling and robust mechanical attachment during testing phase. The current of the terminal block is up to 10A which is valuable for the over current rating test. This also allow modularity where the blocks can be easily configured in different number of positions for design changes.

2.2.2 Wire Harness

The wire harness selection is crucial for routing of electrical signals and power between each subsystem and components within the spacecraft. The unfirm selection of the ADCS is 28 AWG wire based on the current-carrying capacity, mechanical flexibility as well as power drop. The current capacity as shown in the following table is up to 1.4 A which is enough for low power connections between the motors as well as magnetometer breakout board. 28AWG offers thinner wires which lead to weight reduction. The signal integrity of high frequency and twisted pair technique of 28AWG is also implemented to mitigate EMI and crosstalk.

The wire is insulated by PTFE for space graded environment with the specification of 10x cable diameter.

Table 2.8. Copper wire specifications.

Parameter	Value
Conductor Material	Copper [Cu]
Conductor Diameter Cross sectional Area	0.32mm 0.08mm^2
Resistance @20C	212 Ohm/km
Current Carrying Capacity	1.4A
Voltage Rating	300V
Insulation Material	PTFE

Operating Temp	-65C to 200C

2.3 PCB Design

2.3.1 Ventura ADCS Board

Following the architecture, a schematic was developed in KiCad 8.0 to capture the interface between the microcontroller as well as all of the peripherals and components to make up the ADCS Ventura board. The Ventura Board has a dimension of 95.2 mm x 89.0 mm. This is the formfactor of TechEdSat bus requirement. Therefore, all of the components being placed with the board must comply with the requirements.

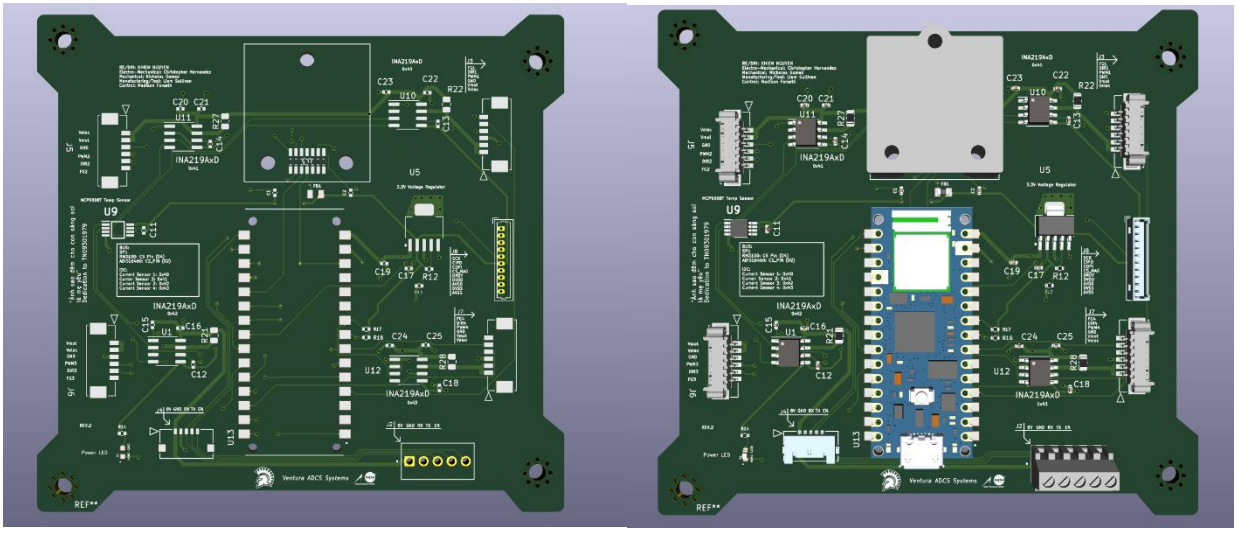


Figure 2.7. PCB populated component (left) and unpopulated (right).

In Figure 2.7, the PCB is designed with the copper path separation width of 0.2 mm and pad with of 0.5 mm for the design of manufacturing to mitigate the appearance of short circuits. The PCB is made by four layers which allow for isolation of signal lines such that I2C and SPI communication lines are on different copper plane. The choice of 4 layer PCB board also help dissipate heat by incorporating thermal vias across the board. The PCB is also made of the FR-4 materials between two thing later of copper which has the woven glass reinforced epoxy resin. The prepreg layer sandwiched between the cores allow food strength and water/humidity resistance allow them to be a food insulator in PCB application.

After the layout process, a Gebrer files containing ASCII vector files was sent to PCBWay for manufacturing for the production of five PCB unpopulated modules like in the figure below.

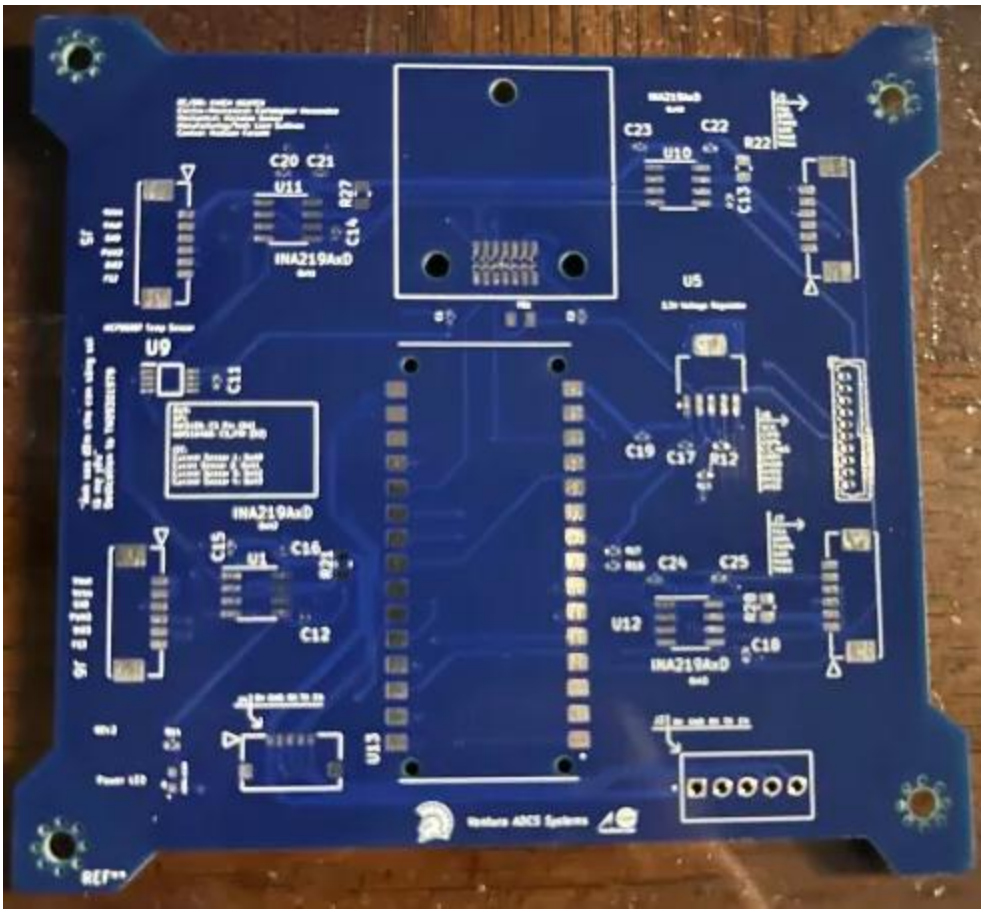


Figure 2.8. Unpopulated PCB.

The board was then manually soldered to create the fully finished board as shown in Figure 2.8.

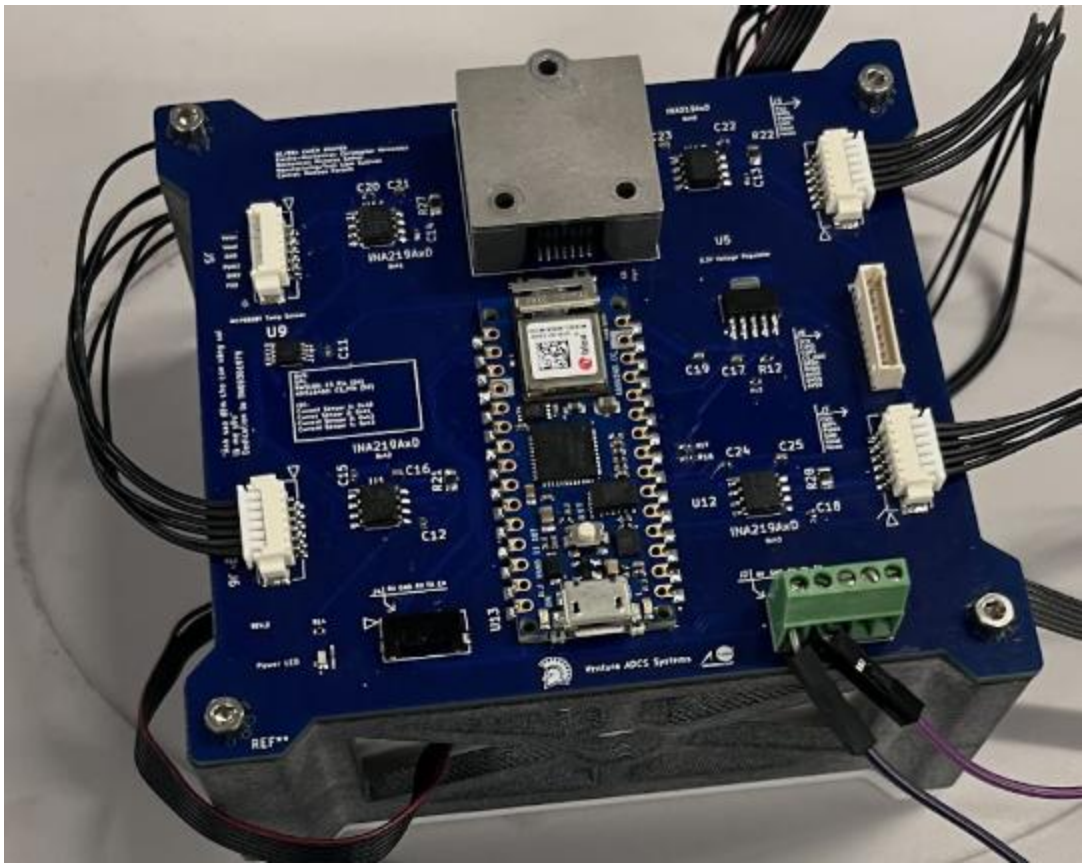


Figure 2.9. Ventura ADCS board.

The Ventura Board random vibration analysis was conducted via Ansys Mechanical. The methodology can be referenced in Figure 2.8. such that the modal analysis was conducted to populate the natural frequencies as an in put for the random vibration analysis.

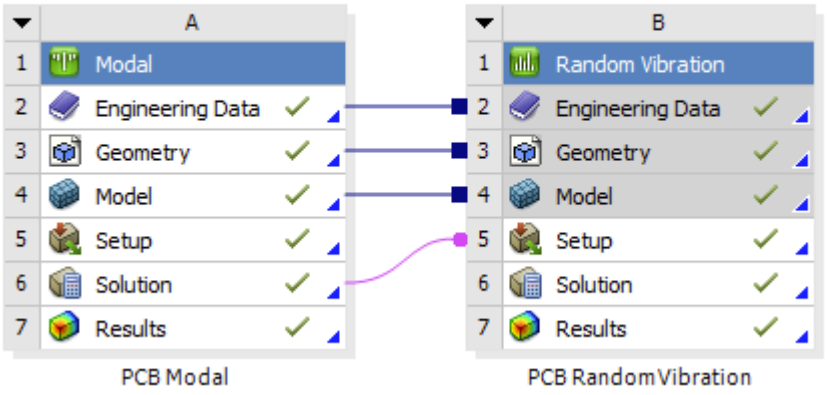


Figure 2.10. Random vibration analysis of Ventura PCB

The materials defined in the PCB are FR4 which is a glass fiber epoxy laminate which is known for its high strength and good electrical insulated and resistance to moisture. This material

is used to bring down the cost of hardware but does not compromise the reliability of the electronics systems.

For the geometry this is a thin 2mm four layers board with the dimensions of 92.5mm x 88.9mm in which the platform resides at the structural center and the mounting PCB attachment would slide within the Cubesat holding/mating rails.

The constraint of the PCB body would be at the 4 mounting holes to simulate the natural frequency as well as acceleration mode.

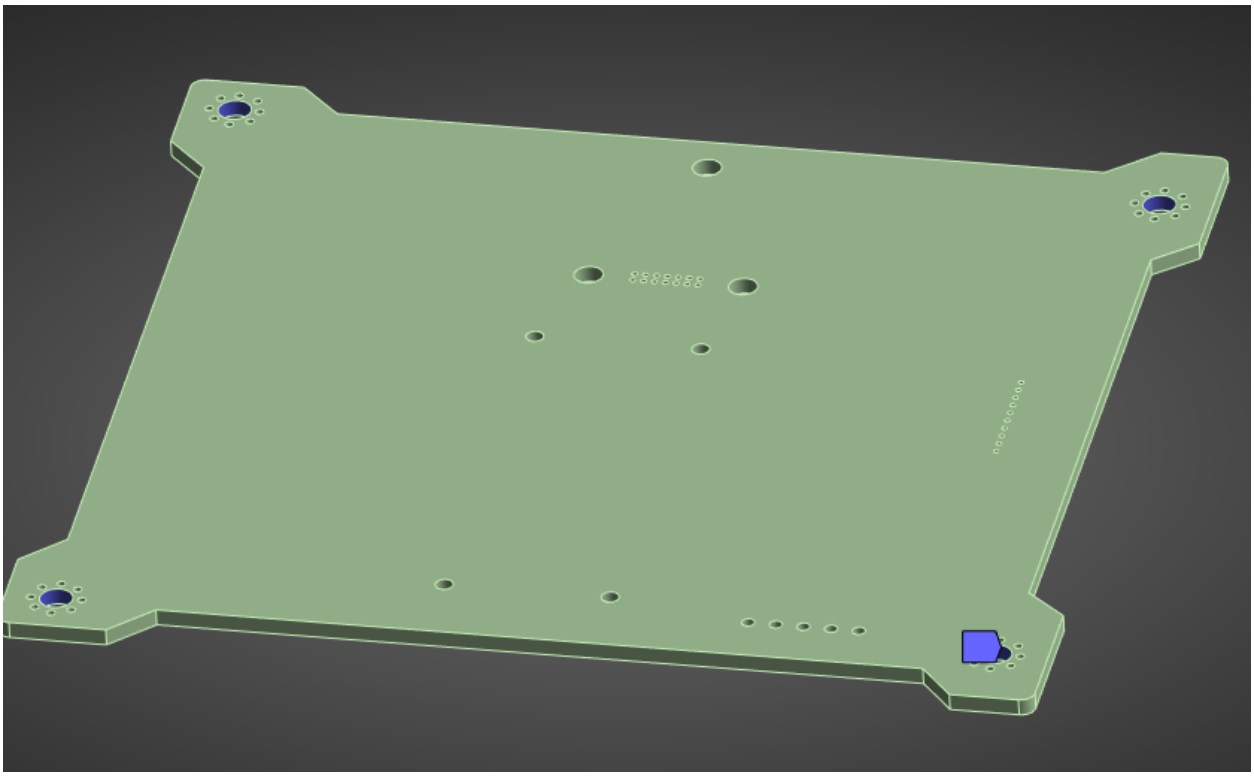


Figure 2.11. PCB constraints at four mounting holes.

The boundary condition consist of a Power Spectral Density acceleration load input profile shown in Table 2.9. and Figure 2.12. The entire profile sweep from 20 to 2000 Hz. The rising and falling slopes of the profile are from 20 to 80 z which are the power spectral density ramps up. Once the peak is reached, which is around 350 Hz it decreases down to 0.02 by 2000Hz domain.

Table 2.9. PSD Profile.

	Frequency [Hz]	Acceleration [(m/s²)²/Hz]
1	20.	3.e-002
2	80.	0.2
3	350.	0.2
4	2000.	2.e-002
*		

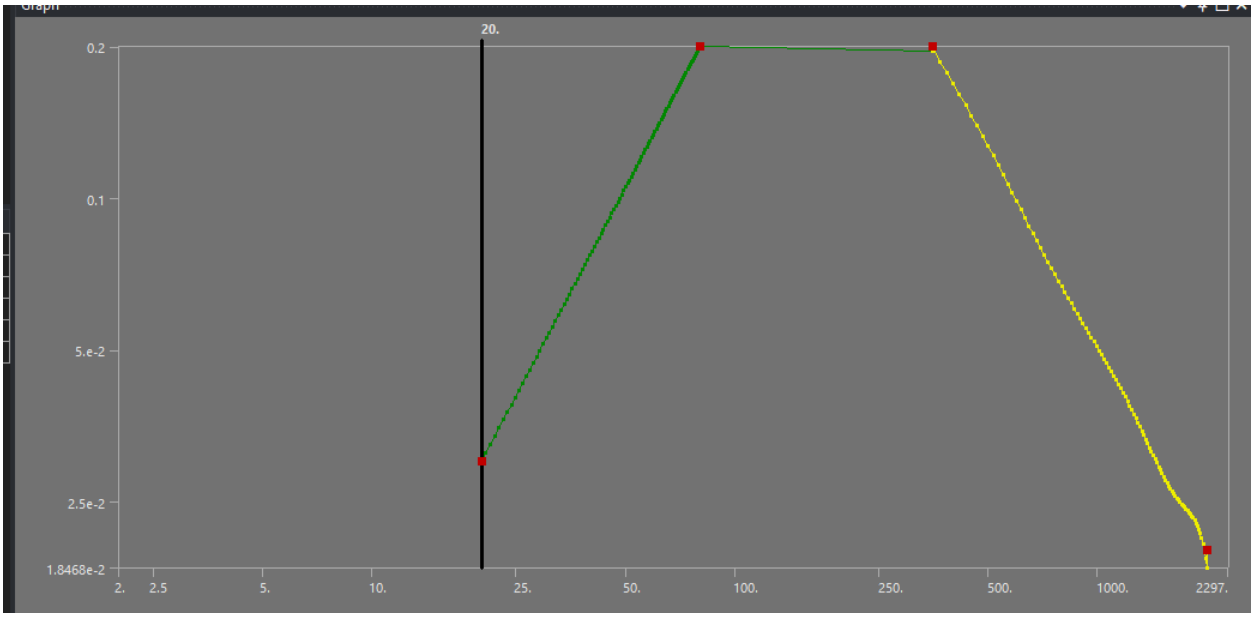


Figure 2.12. Graph of the PSD profile.

In Figure 2.13 10 modes of natural frequencies of the modal analysis re extracted which range from 583 Hz to 4000 Hz. The first mode which is at 538 Hz shows the out of plane bending in the Z direction in which there is a global bending located directly at the center in the Z direction. Since it is at the lowest frequency it is to be concluded that it can be easily excited in a random vibration frequency range. For the second mode of 1347 Hz there are two local lobes at the positive and negative side quadrant near the PCB connectors as well as the IMU location to be shown. This help to inform PCB design and components placements to be considered. The last mode which is near the 2000 Hz random vibration would be at 2301 Hz. This is essentially 300 Hz above the excitation of the random vibration where there are presences of multi-lobes in all 4 quadrants. This mode can be very concern for local stress and concerning features of the design constraints.

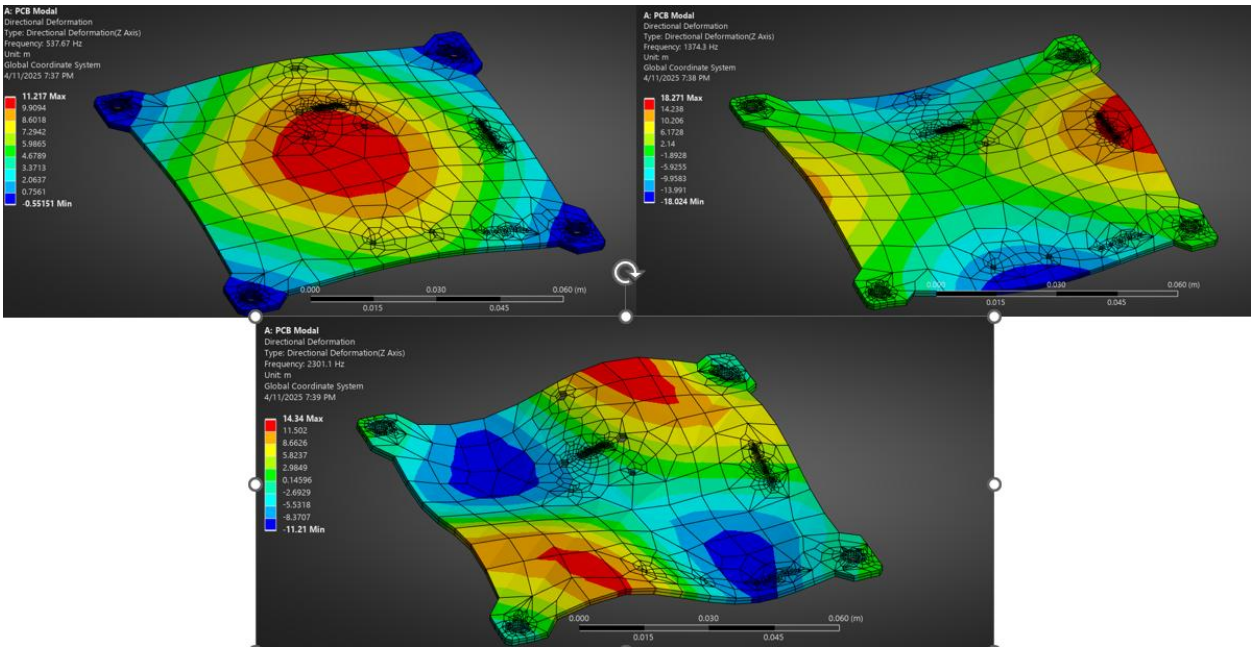


Table 2.13. PCB modal analysis mode.

In Figure 2.14 shows the directional displacement of the random vibration results of the PCB. The interesting point in this analysis are the tree holes mounting features for the IMU where the legend shows that the displacement is around 9 micrometers at 1 sigma probability which is 68%. This means that at 68 percent of the time the instantaneous displacement due to random load will occur within that magnitude. The 3-sigma regime shows that there are 27 microns around the z ais. This means that the IMU still experiences very small displacement of under random vibration load. However, since the system is carrying a very sensitive MEMS IMU, it would mean that we need to avoid the center mounting as much as possible to reduce the deflection which might induce solder fatigue.

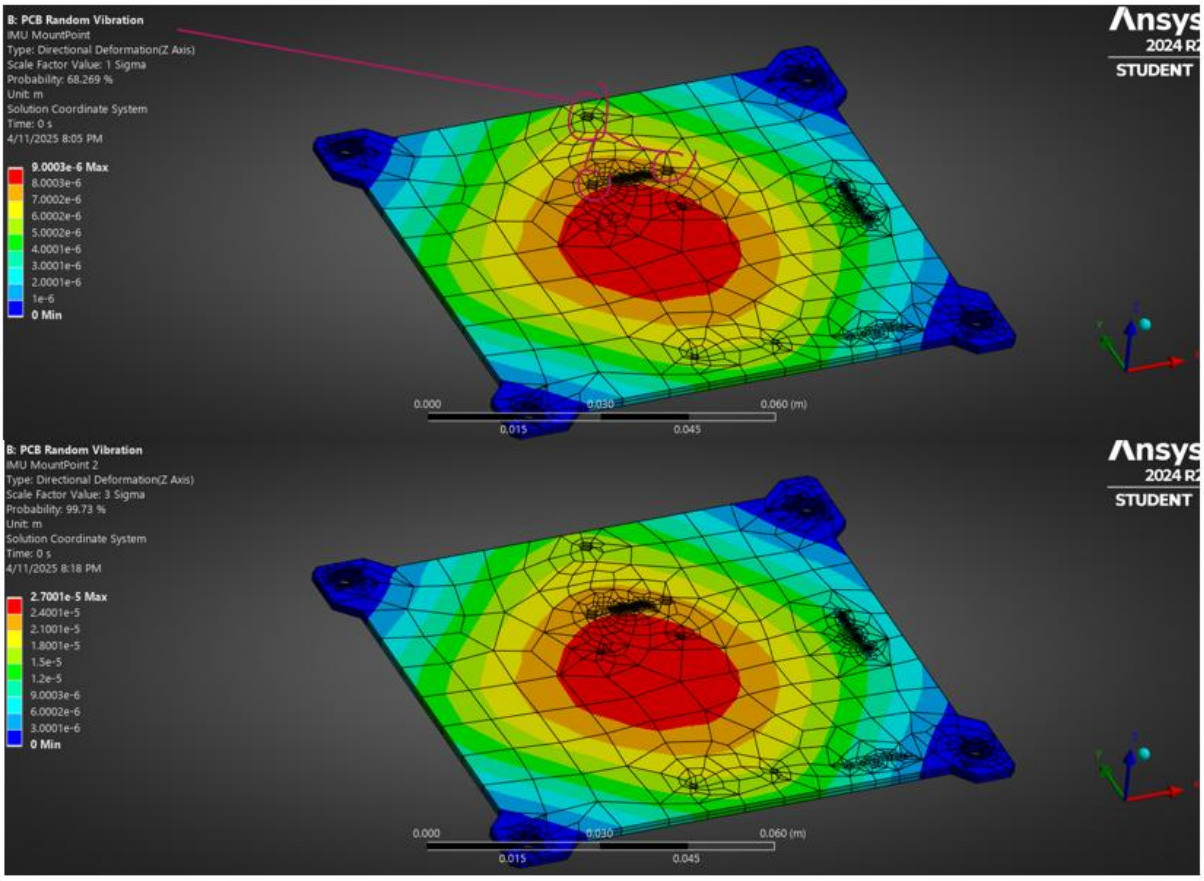


Figure 2.14 PCB random vibration analysis.

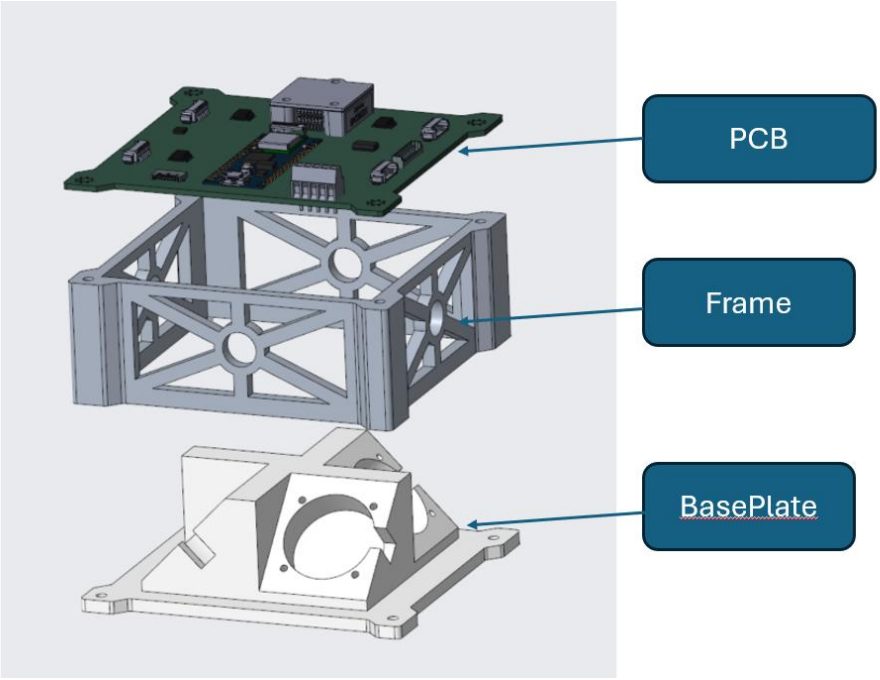


Figure 2.15. Final assembly stack up.

Figure 2.15 shows the entire stackup of the Ventura ADCS systems. The assembly is used with the rods mounted techniques where there would be a shoulder screw to run through the entire assembly and hex nut to lock in at least two threads at the bottom of the assembly.

2.3.2 Sun Sensor Board

Similar to the Ventura ADCS board, the sun sensor board was made in KiCad 8.0. However, the sun sensor carries very small little photodiodes in the middle which make the breakout board very small which is 22.8 mm x 15.5mm. The requirement of small sun sensor board must be complied with due to the fact that there need to be four of them on each quadrant of the CubeSat for sun sensing operation.

Since the footprint and the board is only digital and does not interface with other aggressor EMI electronics or components, the layout was only compiled of two layers for simplicity and cost saving. There are features of debugging pings and test points for early development and prototype purposes.

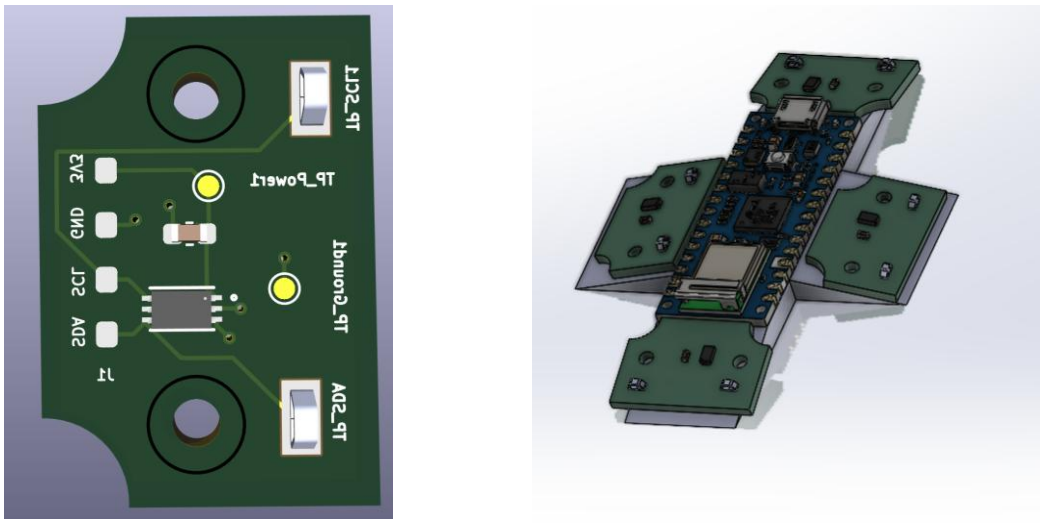


Figure 2.10. Sun sensor board (left) and sun sensor integration (right).

Figure 2.10 shows the individual board on the picture on the left with the sun sensor being in the middle. The picture on the right shows the sun sensor board being mounted in a pyramidal configuration for sun presence monitoring. The concept of sun monitoring confirm that the sun vectors are detected as well as the sun's position respect to the body frame should be where it should be. A set of three or more cosine sun sensors can determine the sun vector where each radius of the circle indicate half cone angle sensed by each sensor. The sun sensor location is intersection of three circle. The error in the knowledge of the sensor max current I0 prevent the circles from intersecting at a single point.

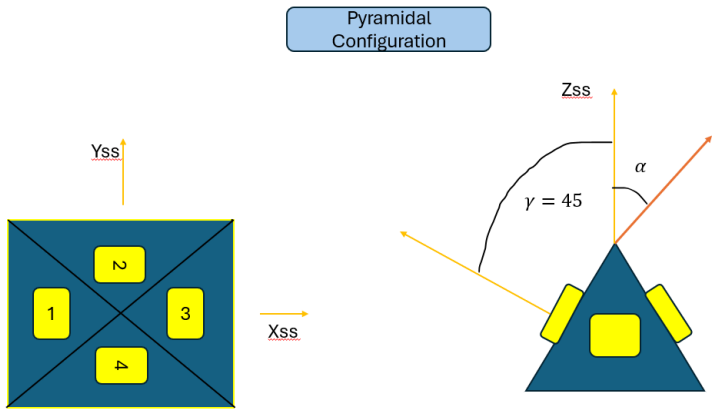


Figure 2.11. Pyramidal configuration (45 degrees).

Figure 2.11 shows the common sun sensor configuration which is the arrangement of four individual cosine sun detectors ells in all faces of the spacecraft to project toward the sun and detect the sun's intensity. The tilted angle of pyramidal configuration is 45 degrees. This provides 2-axis sensing with respect to the sun-line vector. Quadrant one and Quadrant three determines the sun Azimuth angle which is the rotation α about the y-axis in the figure. Quadrant two and four determines sun elevation angle which is rotation , β about the x axis.

The sun azimuth angle α is calculated from quadrant 1 and Quandrant 3 currents such that:

$$(I_1) = I_0 * \cos(45 + \alpha) = I_0[\cos(45)\, cos\alpha - \sin(45)\, \sin\alpha] \quad (2.3)$$

$$(I_3) = I_0 * \cos(45 - \alpha) = I_0[\cos(45)\, cos\alpha + \sin(45)\, \sin\alpha] \quad (2.4)$$

Therefore, the sensor output is constructed as the ration of the current difference divided by the current sun:

$$\frac{I_3 - I_1}{I_3 + I_1} = \frac{I_0 * \sqrt{2} sin\alpha}{I_0 * \sqrt{2} cos\alpha} = \tan(\alpha) \quad (2.5)$$

The same calculation can be applied to quadrant 2 and 4.

3. ADCS Budgeting and Dynamics Modeling

3.1 Attitude Dynamics with Reaction Wheels

The rigid body dynamics is the derivative of angular momentum equal to the torque applied.

$$\tau = \frac{d}{dt}H \tag{3.1}$$

Where H is the angular momentum, and τ is the torque being applied. If the rigid body dynamic which is the CubeSat under no external torques such that $\tau = 0$, it is said to have angular momentum to be constant.

However, the equations of motions for an spacecraft with only reaction wheels as the actuators, the total angular momentum of the satellite in the body frame H_{sat}^{b} to be:

$$H^B = (I_B + I_w)\omega^B + h_w \tag{3.2}$$

Where h_w is the momentum of the wheels relative to the spacecraft body frame.

The derived angular momentum established can be also used from the Euler moment equations such that

$$\frac{d}{dt}(H^B) = \tau^B - \omega^B \, x \, H^B \tag{3.3}$$

Adding in the reaction wheels expression yield

$$\frac{d}{dt}\left((I_B + I_w)\omega^B + h_w\right) = \tau^B - \omega^B \times ((I_B + I_w)\omega^B + h_w) \tag{3.4}$$

The expression can be rearranged to yields

$$(I_B + I_w)\frac{d}{dt}(\omega^B) = \tau^B - \frac{d}{dt}(h_w) - \omega^B \times ((I_B + I_w)\omega^B + h_w) \tag{3.5}$$

The term

$$\omega^B \, x \, ((I_B + I_w)\omega^B + h_w) \tag{3.6}$$

Represents the Coriolis effect, $\frac{d}{dt}(h_w)$ is the net torque on the reaction wheels by the spacecraft body and $-\frac{d}{dt}(h_w)$ is the net torque on the spacecraft body by the reaction wheels. Therefore, the derivatives of the angular velocity can also be derived to be :

$$\frac{d}{dt}(\omega^B) = \frac{1}{(I_B+I_w)}[\tau^B - \frac{d}{dt}(h_w) - \omega^B \, x \, ((I_B + I_w)\omega^B + h_w)] \tag{3.7}$$

The spacecraft dynamics and attitude control method using reaction wheel actuators can be done by Eulers approach or Quaternions approach. The angular velocity and angular momentum equations from the body frame need to be be translate to the quaternion's representation. Quaternions is a better preferable option since they avoid singularities (gimbal lock) which Euler has. Quaternions also is much more computationally efficient for feedback controls.

The quaternion q can be represented as

$$q = [q_0, q_1, q_2, q_3] \tag{3.8}$$

Where q_0 is the scalar component and q_1, $q_{2,}$ and q_3 are vector components.

The quaternion is used to update the angular velocity such that

$$\dot{q} = \frac{1}{2} q \times \Omega(\omega) \tag{3.9}$$

Where $\dot{q}$ is the time derivative of the quaternion rate of change of attitude, and $\Omega(\omega)$ is the quaternion form of the angular velocity $\omega = [\omega_x, \omega_{y,} \omega_z]$.

$$\Omega(\omega) = \begin{bmatrix} 0 & -\omega_x & -\omega_{y,} & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_{y,} \\ \omega_{y,} & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_{y,} & -\omega_x & 0 \end{bmatrix} \tag{3.10}$$

3.2 Momentum Budget

The requirement is to perform an angular position change for all x,y and z axis. Such that the ADCS subsystem must be able to execute a 180 degree slew maneuver. This will constitute the rigid body dynamics and their vector mechanics principals projecting the reaction wheel pyramid configuration o the CubeSat's principal axes.

The reaction wheels subsystems are made up of 4 brushless DC motors Faulhaber 2610 B SC which provides max torque of 6mNm and no-load max speed of 6700 revolution per minute. The reaction wheel made from aluminum 6063 has the moment of inertia around $3.11.34\ x\ 10^{-9} kg * m^2$.

The assumption of most 6U CubeSat's moment of inertia to be about:

$$I^B = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.2 \end{bmatrix} kg * m^2 \tag{3.11}$$

The CubeSat is arranged with the pyramidal configuration where the wheels are angled to 45 degrees angles to the x-y plane. This means that each rection wheel contributes angular momentum to the CubeSat's x y, and z axes. The projection for the reaction wheel's angular momentum to each axis is governed by the following equations:

$$H_{RW} = I_{RW} * \omega_{rw} \tag{3.12}$$

$$H_B = H_{RW} * \cos(45) \tag{3.13}$$

$$H_B = H_{RW} * 0.707 \tag{3.14}$$

This means that each wheel contributes about 70.7% to its total angular momentum of the CubeSat's body axis.

Since the assumptions made to not model external torque disturbance, the symmetric configuration allows for equation contributions along all axes and their angular momentum can be written as follows:

$$H_x = 2 * H_{RW} * 0.707 \tag{3.15}$$

$$H_y = 2 * H_{RW} * 0.707 \tag{3.16}$$

$$H_z = 2 * H_{RW} * 0.707 \tag{3.17}$$

To calculate and estimate the maximum angular velocity and angular acceleration with the information given above the following equations are used.

$$H_{max} = \ I_{RW} * \ \omega_{RW} \tag{3.18}$$

The substitute of the reaction wheel moment of inertia of Irw = $311.34\ x\ 10^{-9} kg * m^2$ and $\omega_{RW} = 6700\ rpm$ yields

$$H_{max} = 2.18\ x\ 10^{-4} kg * m^2/s \tag{3.19}$$

$$I_{Bx} * \ \omega_x = \ 2 * \ H_{RW} * 0.707 \tag{3.20}$$

$$\omega_x = \ \frac{2 * 2.18\ x\ 10^{-4} \ * 0.707}{0.1} = 3.08\ x\ 10^{-3} \frac{rad}{s} = 0.176\ deg/sec \tag{3.21}$$

Substitute the value above into angular acceleration formula as below

$$\alpha_{max} = \ \frac{\tau_{max}}{I_{Bx}} = \ \frac{6\ x\ 10^{-3}}{0.1} = 0.06 \text{ rad/sec} = 4.87 \text{ deg/}s^2 \tag{3.22}$$

Using the calculations above the maneuver profile can then be calculated. Such that the acceleration phase would occur first then constant velocity y phase then deceleration phase to nominal mode.

$$t_{acceleration} = \ \frac{\omega_{max}}{\alpha_{max}} = \frac{0.176}{4.87} \ = 0.036\ seconds \tag{3.23}$$

The angular displacement during acceleration can also be calculated to be

$$\theta_{acc} = \frac{1}{2} * \ \alpha_{max} * \ t_{acc}^2 = \frac{1}{2} * 4.87 * (0.361)^2 = 0.032\ degrees \tag{3.24}$$

The remaining angular displacement for the constant velocity can be shown as

$$\theta_{const} = 180 - 2 * \ \theta_{acc} = 180\ degrees \tag{3.25}$$

Therefore, the total 180 degrees slew maneuver will takes:

$$t_{total} = 2 * \ \ t_{acc} + \ t_{cosntat} \ = 1022.77\ secs = 17\ minutes \tag{3.26}$$

3.3 Environmental Disturbances

In the low earth orbit domain there are four types of disturbances that need to be considered when designing the control systems for the ADCS subsystems. These include aerodynamic drag, solar radiation pressure, gravitational torque and geomagnetic torque.

3.3.1 Aerodynamic Drag

Within the low earth orbit domain, the external torque generated by the number of air particles present is the most significant compared to gravity gradient disturbance. As the altitude of the orbit get closer to the Earth, the magnitude of aerodynamic torque increases. The aerodynamic drag effect is presence for Earth orbits to the 600 km range which is where most CubeSat mission operates in. The equation [25] for aerodynamic torque can be calculated by the following equation:

$$\tau = \frac{1}{2}\rho C_d A V^2 (u_v \text{ x } (C_{pa} - C_m)) \tag{3.27}$$

Where ρ is the atmospheric density, V is the satellite linear velocity, u_v is the satellite unit linear velocity vector, and A is the affected surface area defined by the projected to the plan vector perpendicular to u_v. C_{pa} represents the center of pressure on the CubeSat platform which can be calculated as:

$$C_{pa} = \frac{\int x\, P(x) dx}{\int P(x) dx} \tag{3.28}$$

Where x represents the distance from the CubeSat body reference frame's origin and P is the function of x which represents the pressure magnitude. This takes into the assumptions that P is constant at every single point on the CubeSat's surface area A . This distribution of torque produced by the satellites span is small to the torque output by the reaction wheel systems but not to be negligible.

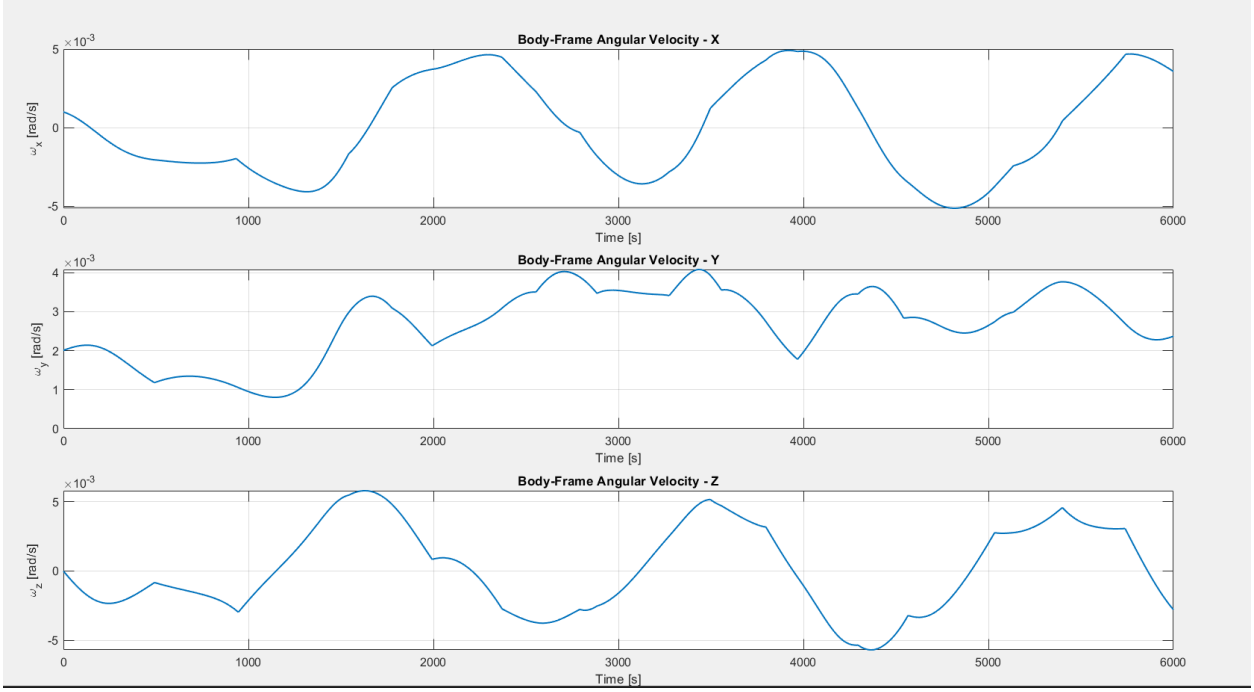


Figure 3.1. Aerodynamic drag simulation.

In Figure 3.1, Matlab was used to compute the simulation of the torque varies over time due to the gravitational torque in low earth orbit effect. The spacecraft had a initial condition of 6 rad/sec on all principal spin axes and the as the aerodynamic torque acts on it changes the angular velocity very minute. Since draft is a dissipative force, the expectation would be for the spacecraft to slowly dampen out he disturbance.

3.3.2 Solar Radiation Pressure

The solar radiation pressure comes from the Sun angle of radiation beam incident on the spacecraft that cause the torque about the spacecraft's mass properties. This would also couple into the albedo effect where the reflection of the solar radiation from the Sun and the infrared emission from Earth body affect the spacecraft's torque [26]. The albedo effect can simply be predicted with advanced simulation tools built from MATLAB [27] or NASA TOMS project to predict the compensation torque and sun sensor modeling. Light has momentum such that the change in momentum generates a radiation gradient pressure on the spacecraft that depends on geometry and optical surface properties. The offset between the center of solar pressure and the center of mass results in the solar torque.

The NASA's Space Vehicle Design Criteria provides an estimation on the worst-case scenario of the radiation force that illuminated on the surface area of the CubeSat. The assumption is that all the incident radiation operates on absorption, reflection and diffusion with a negligible scattering mode such that:

$$dF_{SRP} = \frac{I_{SRP}}{c}\left\{-\left[(1+c_{rs})\cos(\alpha)+\frac{2}{3}c_{rd}\right]\hat{n}+(1-c_{rs})\sin(\alpha)\,\hat{s}\right\}\cos\alpha\;\;dA \tag{3.29}$$

Where I_{SRP} is the energy flux per unit of time through a cross sectional area., c is the speed of light in vacuum, c_{rd} is the coefficient of diffusion, c_{rs} is the coefficient of the diffuses reflection, n is the vector unit perpendicular to the incident surface dA, s is the unit vector perpendicular to n vector of the incident ray. α is the angle between the incident ray and vector n in Figure [3.1].

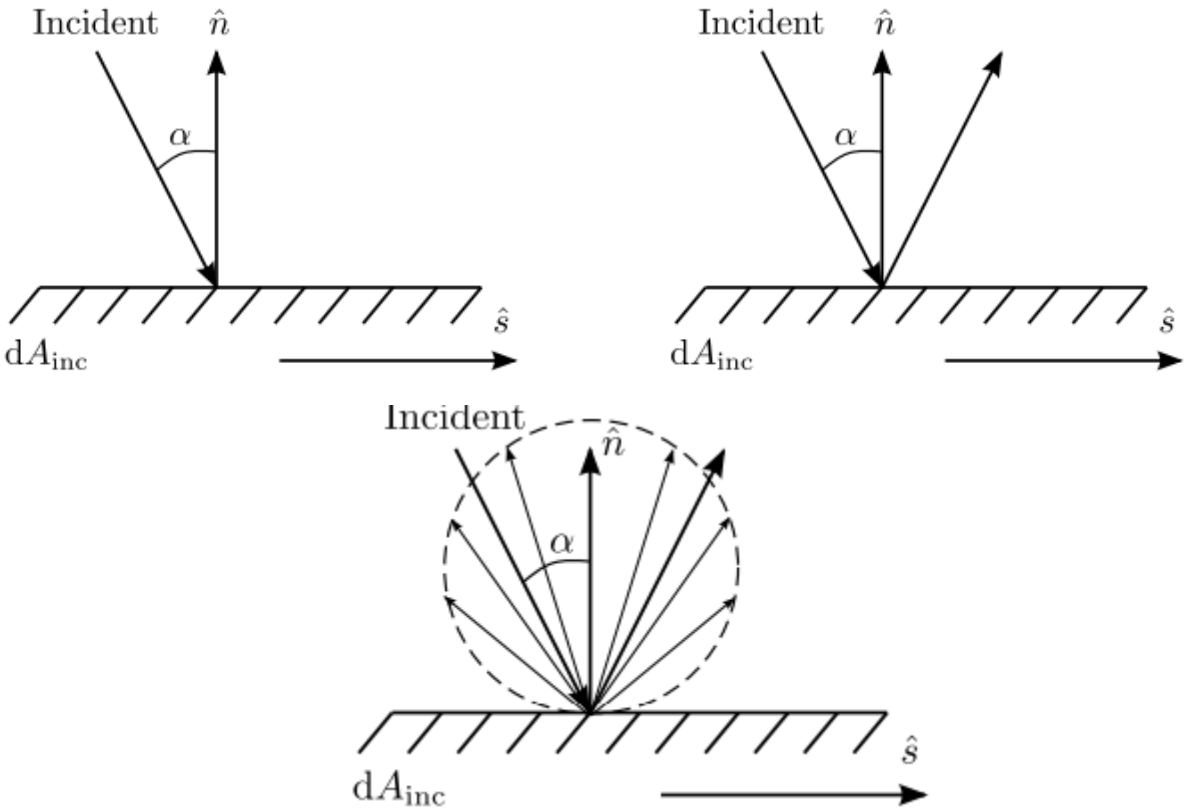


Figure 3.2. Absorption, reflection and diffusion mode on a surface area [28].

Taking the integral on the equation (3.3.2.1) yields:

$$\tau_{SRP} = \frac{F_s}{c} A_s (1 + q) \cos(\alpha) * (u_s C_{pa}) \tag{3.30}$$

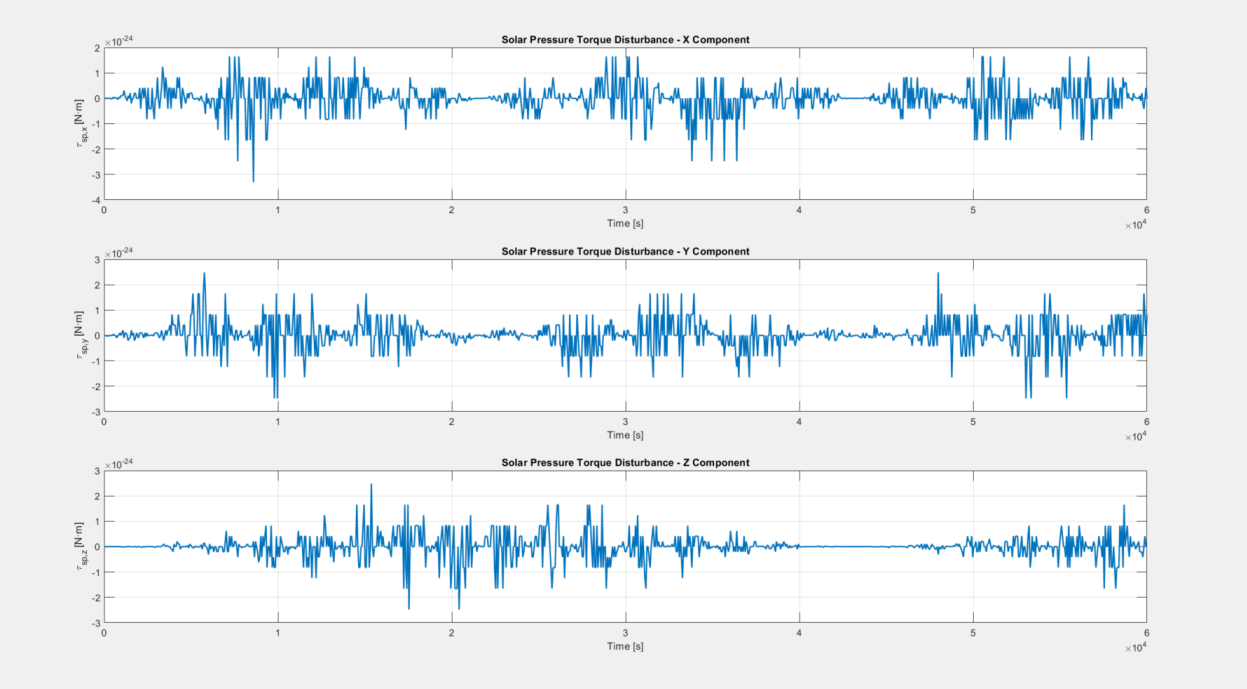


Figure 3.3. Solar disturbance plot.

Using MATLAB script where a simulation of torque vs time can be simulated by the sun where the inputs are the spacecraft center of mass and reflectance factor. There is also a definition of how fast the spacecraft orate and how the Sun's position chances. For the algorithm, at each time step the current rotation angle and the rotation matrix are computed. The sun vector got updated in the orbit frame. Then use a transformation matrix of sun vector ot convert the Sun vector into the specified spacecraft body frame. Then the computed surface area was computed to see how each face the solar force would be exerted by taking the cross product of the force with the lever arm. In this simulation, the spacecraft is rotating around the Z axis for nadir pointing.

3.3.3 Gravitational Torque

All and any non-symmetrical object with rigid body dynamics are subject to gravitational torque due to the variance in Earth's gravitational force on the object. The inverse square law rule is a contributor to the gravity gradient torque in a constant gravitational field. Newton's law dictate that the gravitational force decreases with the square root of 1/r^2 [29]. Therefore, the gravitational torque can be calculated as:

$$\tau_g = \left(\frac{3G_{SRP}}{||r||^3}\right)[r \times (I_{sat} * r)] = \left(\frac{3\mu E_{earh}}{||r||^3}\right)[r \times (I_{sat} * r)] \tag{3.31}$$

Where μ_{earh} is the Earth gravitational constant, r is the position of the satellite center of mass with respect to the earth center of mass, and I_{sat} is the satellite's moment of inertia three by three matrix.

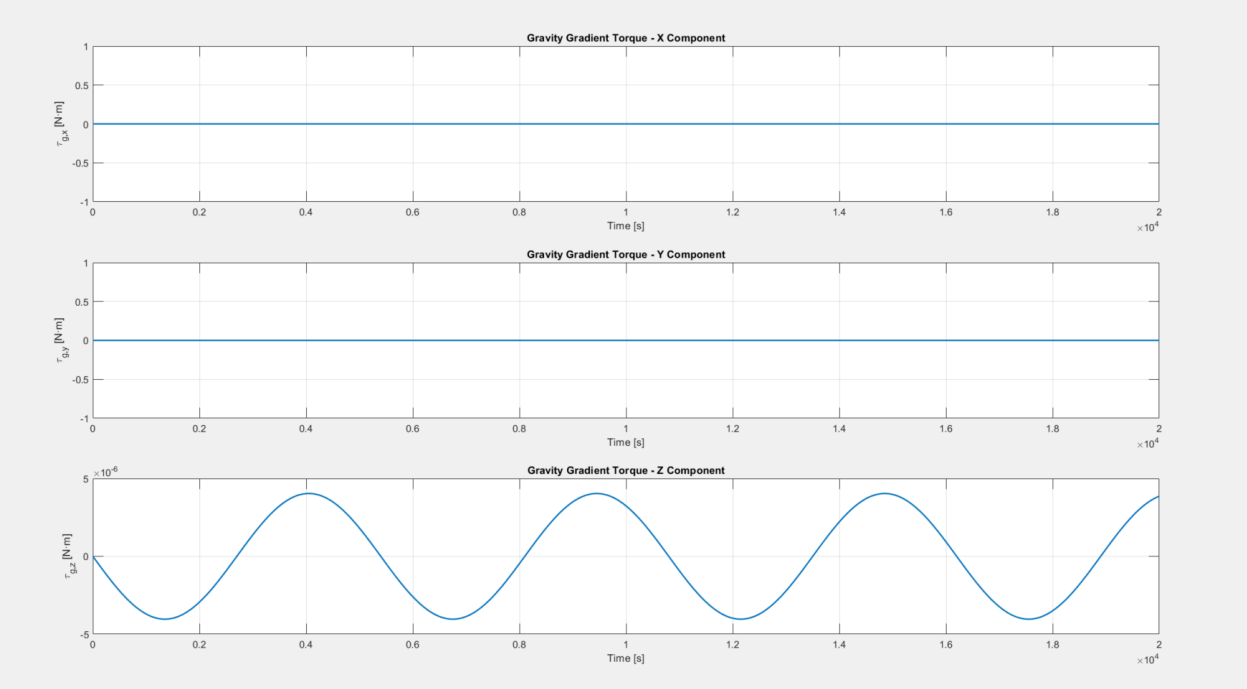


Figure 3.4 Gravitational torque simulation.

As observed in Figure 3.4 There is a sinusoidal wave in the component Z spin axis where the period of these oscillations is related to the spin rate as well las the inertia axe. There is also a torque component cross zero where the vector position liens up in a way for the cross product to become significantly small approaching zero. The amplitude can be seen to be affected by the orbital rate as well as the principal moment of inertia of the spacecraft. The gravity gradient torque shows the align the spacecraft so that the largest moment of inertia points away from Earth .

3.3.4 Geomagnetic Torque and Residual Magnetic Torque

In the spacecraft, all of the electronics that have current and voltage pass through their transmission line or traces on the PCB produce magnetic dipole that interacts with the Earth's magnetic field generating torque disturbances. Other source of magnetic field can also come from eddy currents and hysteresis of the magnetorquer. The model of the Earth; s geomagnetic field can be found by Wertz such that:

$$B_{Earth} = \left(\frac{R_{Earth}^3 * H_0}{\mathrm{r}^3} \right) [3(m_{north} * \mathrm{r})\mathrm{r} - m_{north}] \tag{3.32}$$

Where $R_{Earth}^3 * H_0$ represents the strength of the geomatic moment and r is the position of the satellite the Earth coordinate frame reference frame, m_{north} is the normal vector of the position of the geomagnetic pole and Rearth is the radius of Earth.

The torque created by the residual magnetic moment is:

$$\tau_{risidual} = M_{sat} \times B_{Earth} \tag{3.33}$$

Where M_{sat} is the satellite local magnetic induction and B_{Earth} is the Earth magnetic field. This applies ot permanent magnets, electromagnets or any closed current loop.

One of the main benefits for operating in low earth orbit with the induced geomagnetic torque is that there is a redundancy introduction of a magnetorquer subsystem using magnetic to dump momentum. This allows minimal interference with mission ops and has much lower mass tan thruster-based systems. However, the disadvantage is that it trap electron belts.

3.4 Sensor Modeling

3.4.1 Gyroscope Modeling

In Fundamentals of Spacecraft Attitude Determination and Control, the gyroscope measurement model can be written as follow [21]:

$$\omega(t) = (1 + A)\omega^{true}(t) + \beta^{true}(t) + \eta_v \tag{3.34}$$

Where ω is the observed and measured angular velocity in the gyroscope reference frame, ω^{true} is the real angular velocity within the same corresponding frame, A is the misalignment rotation matrix, I is the identity matrix, β^{true} represent the drifting bias of the gyroscope, η_v represents the independent zero mean Gaussian white added noise. This can also be used for scale factors and misalignments.

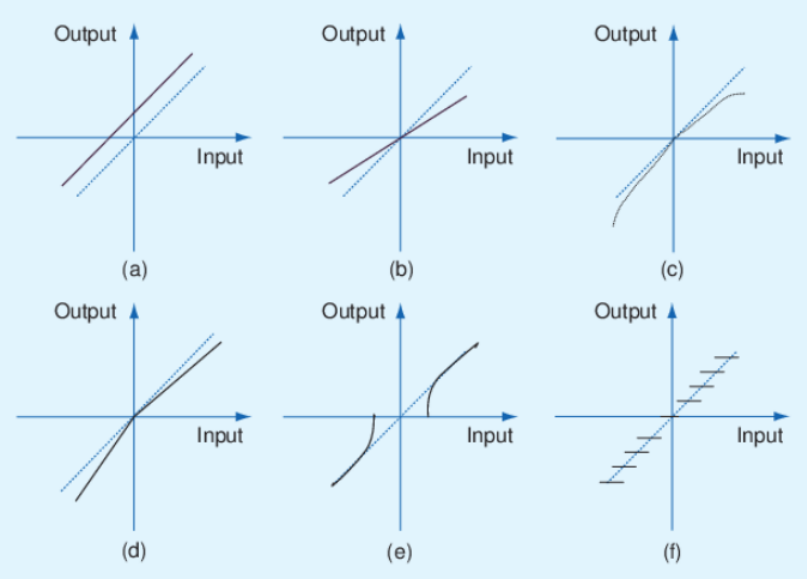


Figure 3.5. Common gyro input/Output error types [22].

In Figure 3.5, graph (a) shoes the bias error which outline the shifted of the output line parallel to the input axis indicating that there is a consistent offset. Thid occurs when the gyroscope produces a non-zero output respect to the zero input. Graph(b) shoes the a response with scale factor error in which the output slope start to drift or deviate from the ideal condition. A nominal response should have a proportional correlation between the input and output. Graph (c) shows the nonlinearity error such that the output function response deviates from a straight line after certain ideal inputs response. This is due to sensor manufacturing imperfection or design error. Graph (d) shows the asymmetry error which leads to the uneven response on the angular velocities rate in the true opposite direction. Graph (e) shows the hysteresis error response with a "dead-zone". This happens due to the current input and previous input history are interfering with the output. Graph (f) represents the quantization error which acts a s a step function instead of a continuous ideal line. This occurs when there is a limitations of digital resolution where digital gyroscope have quantized into finite steps.

The bias instability and white Gaussian noise can be presented by the Markley methods. The first component of the bias instability (Random Walk Noise) came from the slow variations and differences in the gyroscope's bias over time due to the architecture of the sensor. The model can be built on a bias at step k+1 depends on the previous bias k with additional small random perturbation which can be shown in this equation below. The symbol σ_n reperesnet the bias instability standard deviation, and N(0,1) is the standard simulated Gaussian random walk variable noise.

$$bias_{new} = old_{bias} + \sigma_n\sqrt{dt} * N(0,1) \tag{3.35}$$

The white noise is essentially a high frequency noise arising from the electronic disturbances n the sensor. This is dictated by a normal distribution and is independent of the previous values. The noise of the signal can be modeled by the equation below where σ_u^2 is the white noise standard deviation.

$$noise_{total} = 0.5 * (bias_{old} + bias_{new}) + \sqrt{\frac{\sigma_v^2}{dt} + \frac{\sigma_u^2 * dt}{12}} * N(0,1) \tag{3.36}$$

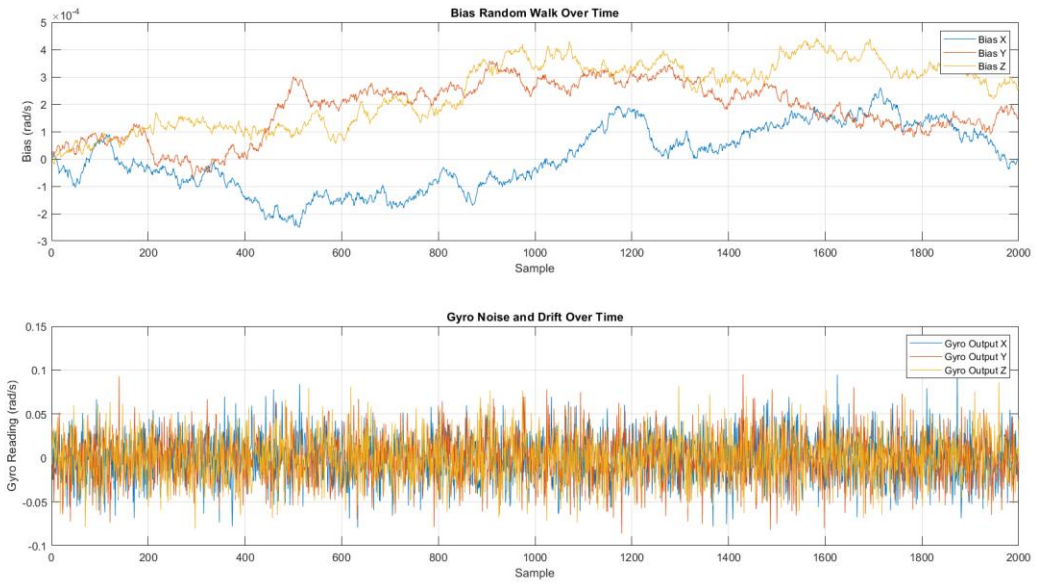


Figure 3.6. Bias random walk over time plot (top). Gyro noise and drift over time plot (bottom).

The figure 3.6 illustrates the bias random walk for all three spin axes overtime. This is a stochastic process vehicle where the sensor bias drift appears to be slow and smooth which is a characteristic of medium to high frequency bias instability. The amplitude of the drift remains within the specs of the operation baseline. However, as time goes on the correlation of the bias noise is being stretched out which mean that long term data might not be accurate but short-term data will be accurate.

The noise is the Gaussian distributed variation around the bias mean value. Such noise can be reduced and mitigated by high pass filtering to reduce the fluctuation. The white noise also seem to exhibit a flat power spectrum where the bias instability contributes to 1/f noise. The Allan Variance analysis curve can be used to demonstrate the noise sources where the white noise is dominant at short term timescale while bias instability is an important metric for long term timescale.

3.4.2 Sun Sensor Modeling

The spacecraft is hosting 6 sensors where each of the six sensors needs to be modeled separately. This means that each of the sensors would have six framed to be defined. Out of the six, four frames are defined by the rotation of body frame around the Y axis and the other two frames are around the Z axis. Albedo radiation is also assumed to come from the nadir such that the vector is acquired in the body frame. This is doable by allowing the rotation of satellite position vector in the ECI frame with the ECI to body quaternion. The current induced within the

coarse sun sensors is again to be proportional to the cosine of the angle between the vector perpendicular to the sensor and the incident light presented in 3.3.5.1.

$$cos(\theta) = Sensor * [1\ 0\ 0]^T \tag{3.37}$$

In the case that $cos(\theta)$ is lesss than 0, the sensor does not detect any light from the source.

As for Albedo affect the currents of each sensor are calculated by summing up the currents of each sensor collected by sun rays and Earth Albedo. However, the albedo percentage must be scaled due to the Poisson noise.

$$i_{sun} = Sensor * [1\ 0\ 0]^T \tag{3.38}$$

$$i_{sun} = albedoSensor * [1\ 0\ 0]^T \tag{3.39}$$

$$i_{total} = i_{sun} + i_{albedo}\ [1 + 0.01w_k] \tag{3.40}$$

To obtain the sun vector the following equation can be used where the currents in quadrants pair (opposite) can be subtracted to get the normalized vector in the body frame:

$$Sun_{vec} = \frac{(i_1 - i_4, i_6 - i_5, i_2 - i_3)}{\left\|(i_1 - i_4, i_6 - i_5, i_2 - i_3)\right\|} \tag{3.41}$$

The digital coarse sun sensor noise modeling contains the electronic noise, albedo effect, quantization noise and poison noise. Electronics noise comes from the random fluctuation in the sensor's current output due to thermal effects and electronics. The Albedo Effect comes from the unwanted reflections from Earth's surface by the Sun 'rays affecting its readings. Since this is a digital sensor, quantization noise is heavily affected on the analog to digital conversation processing. Poison noise is essentially a variation in the number of detected photons of the sin sensor.

Using a MATLAB script a simulation was set up from 0 to 6000 seconds with a time step of 10 seconds for sun-position vector extraction modeling. The orbit assumption is to be a simple Low Earth Circular orbit in the equatorial plane. The sun position is computed from the Sun's ECI coordinate in each of the time step. The Albedo script inject noise into Earth reflects sunlight in the sensor's field of view this can be show in a tiny shift of near day-night transition. The periodic variation can be seen in Figure 3.5 with sinusoidal behavior across 3 spin axes. The X-spin axis of the sun vector has more significant impact of fluctuations compared to the other two axes, On the other hand, the Posin noise causes jitter in the sun estimate vector such can be seen in Figure 3.5. This clearly shows the need to apply Kalman Filtering to estimate and correct for the Sun Vector deviations.

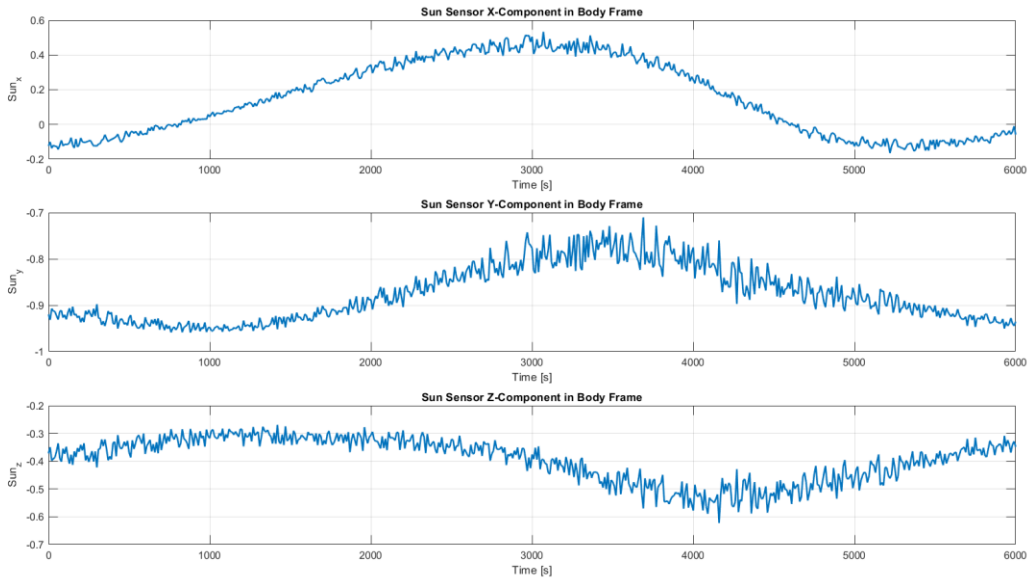


Figure 3.7. Sun sensor simulation with noises.

3.5 Open Loop Monte Carlo Modeling

Using the equation 3.1.7 with addition of disturbance torque yields the following equation:

$$\frac{d}{dt}(\omega^B) = I^{-1}[-w(t)x\,(Iw + H_{rw}) + \tau_{external,disturbance}(t) + \tau_{sat} \tag{3.42}$$

Where $w(t)$ is the initial satellite body frame angular velocity I is the satellite moment of inertia, H is the reaction wheel angular momentum, $\tau_{external,disturbance}$ is external disturbances torque due to gravity, aerodynamics, and geomagnetic disturbances, and τ_{sat} is satellite control torque.

Using the constants in the following table

Table 3.1 Hardware constants

Parameter	Value	Description
I_{sat}	$[1, 1, 0.4]\ kg * m^2$	Moment of inertia of the satellite
$I_{xx}(Satellite)$	1	Satellite moment of inertia about the x-axis

$I_{yy}(Satellite)$	1	Satellite moment of inertia about the y-axis
$I_{zz}(Satellite)$	0.4	Satellite moment of inertia about the z-axis
$I_{reactionwheel}$	diag([1.11×10−7,2.04×10−7,1.11×10−7]) $kg * m^2$	Moment of inertia of the reaction wheels (diagonal matrix)
$I_{xx}(Reaction\ Wheel)$	1.11×10−7 $kg * m^2$	Reaction wheel moment of inertia about the x-axis
$I_{yy}(Reaction\ Wheel)$	2.04×10−7 $kg * m^2$	Reaction wheel moment of inertia about the y-axis
$I_{zz}(Reaction\ Wheel)$	1.11×10−7 $kg * m^2$	Reaction wheel moment of inertia about the z-axis

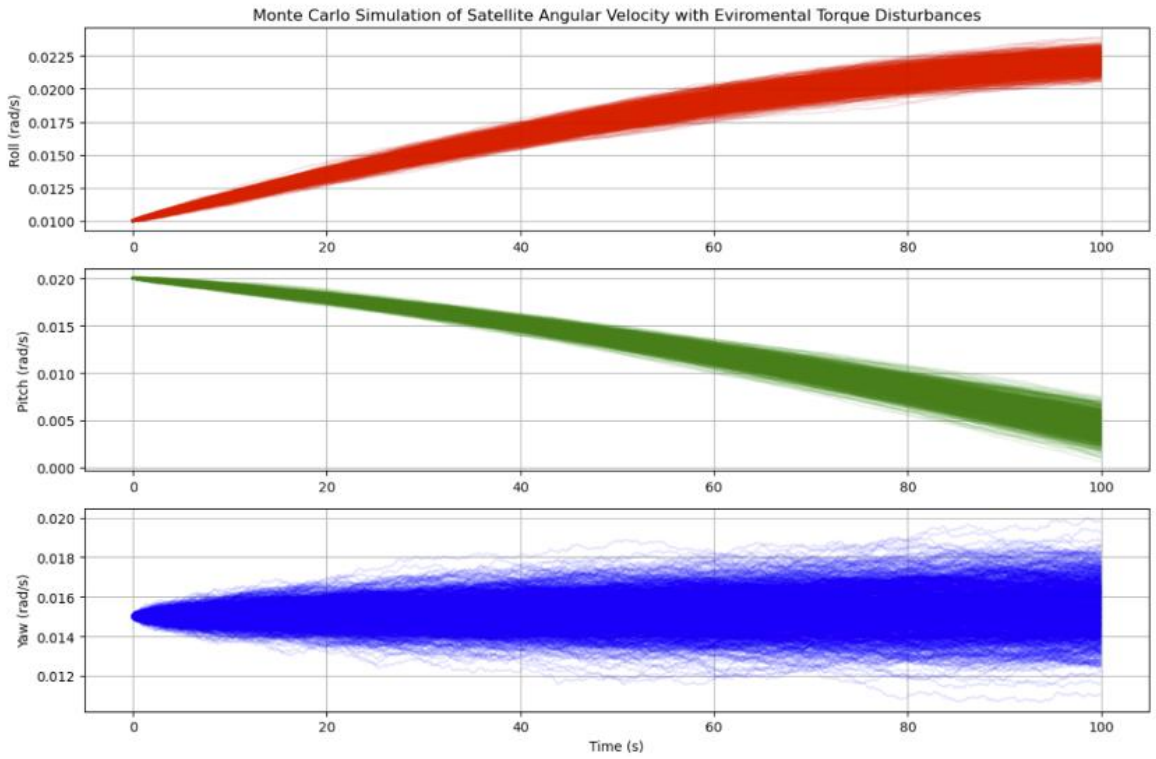


Figure 3.8. Monte Carlo simulation of satellite angular velocity with environmental torque disturbances (open loop).

The angular velocity drift was observed in 1000 runs of simulation for all 3 axes as time increase, this is due to the absence of control torque algorithm where the environmental disturbance cause satellite to accumulate randomized angular momentum reading to angular velocity drift. There is great variability across the simulation where increasing number of runs will have more uncertainty. The biggest factor that can be observed lays in the dominance of magnetic disturbances since this in low earth orbit.

As for the trend, the roll angular velocity increase spread and magnitude over time this could be that the roll axis mass property is coupled to certain disturbances . For the pitch angular velocity, it is very similar to roll. For yaw angular velocity, this shows significant variability which can be explained by the satellite's asymmetrical inertia. Since the lower moment of inertia in yaw axis lead to much faster angular acceleration.

4. Motor Modeling and Control Planning

4.1 Motor Wheel Modeling

DC and BLDC motors are typically modeled by electro-mechanical concepts that can be shown in Figure 4.1.

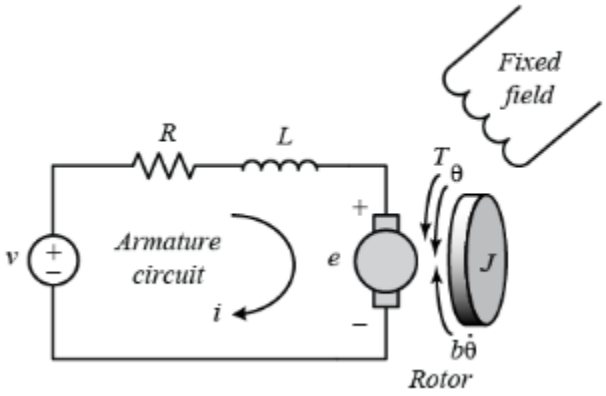


Figure 4.1. DC motor modeling [30].

From Figure 4.1 the amateur circuit is used to model the DC motor such that the passive elements of resistor and induct are lumped element for the resistance and inductance characteristic of the harnessing and the wire within the motor. Lowercase e represents the back EMF force generated by a motor which opposes the input applied voltage. The magnetic field interacts with the motor's permanent magnet or field winding then cause the rotor to spin. As the rotor of shaft is spinning the winding cuts through the B field of the magnetic flix line which induce a voltage. This induced in voltage which is opposite of the polarity y to the applied input voltage therefore called by EMF.

On the mechanical side the motor's rotational motion {K} with damping {b} represents the friction or resistive force to the dynamic motion. To derive the transfer function of the model, torque equations back emf and Kirchhoff's voltage law are used.

Torque Equation in term of current and torque constant of the motor

$$\tau \quad = K_t \times \mathrm{i} \tag{4.1}$$

Back EMF force can be represented the angular velocity of shaft and the motor torque constant in eq. 4.2:

$$e = K_i \times \dot{\theta} \tag{4.2}$$

Equations of motor dynamics are represented by the rotor inertia, the angular acceleration, angular velocity and the motor damping torque.

$$\mathrm{J}\ddot{\theta} + \mathrm{b}\ddot{\theta} = K \times \mathrm{i} \tag{4.3}$$

To solve for the circuit using the Kirchoff Voltage law which yield a relationship of V which is the input voltage, Ldi/dt is the voltage across the inductor, Ri is the voltage drop across the resistance and the is back emf in the equation 4.4.

$$L\frac{di}{dt} + Ri = V - K\dot{\theta} \tag{4.4}$$

After applying the Laplace transform of the dynamics and the electrical models yield equations 4.5 and 4.6 in term of variable s.

$$s((Js + b)\Theta(s) = KI(s) \tag{4.5}$$

$$(Ls + R)Is((s) = V(s) - Ks\Theta(s) \tag{4.6}$$

Combining 4.5 and 4.6 would yield the open loop equation 4.7 where the relation is to evaluate the angular velocity per the applied input voltage.

$$\frac{\dot{\Theta}(s)}{V(s)} = \frac{K}{Ks + b)Ls + R) + K^2} \tag{4.7}$$

For the Faulhaber 2610 BLDC motor such constant can be used to evaluate the open loop performance of the motor in table 4.1.

The transfer function can also be further expressed in start space form in equations 4.8 and 4.9 where the output are the rotational angular velocity.

$$\frac{d}{dt}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} = \begin{bmatrix}\frac{-b}{J} & \frac{K}{J}\\ \frac{-K}{L} & \frac{-R}{L}\end{bmatrix}\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{L}\end{bmatrix}V \tag{4.8}$$

$$y = [1 \quad 0\,]\begin{bmatrix}\dot{\theta}\\ i\end{bmatrix} \tag{4.9}$$

Table 4.1. Parameters of FB2610 Motor

Symbol	Description	Value
Js	Moment of inertia of the rotor	0.01 kg·m^^2
b	Motor viscous friction constant	0.1 N·m·s
Ke	Electromotive force constant	5 V/rad/sec
Kt	Motor torque constant	8.8 mN·m/Amp
R	Electric resistance	1 Ohm
L	Electric inductance	0.5 H

Using MATLAB to model the state space equations as well as table 4.1 the Figure 4.3 and Figure 4.4 represent the step response of the motor open loop.

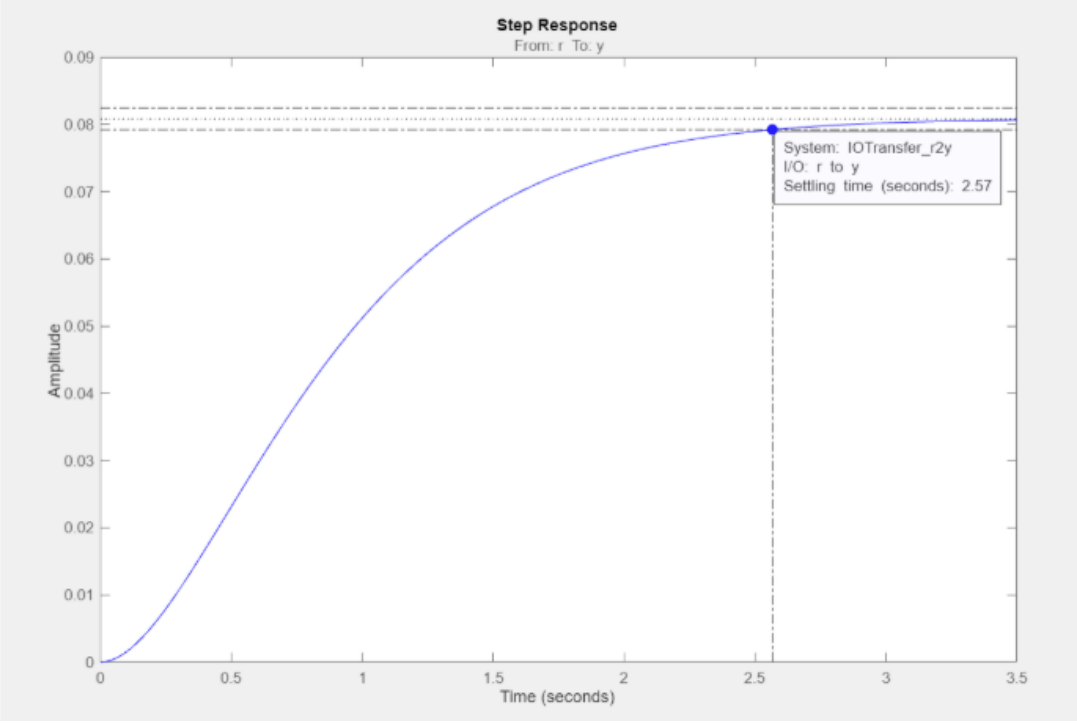


Figure 4.2 Open loop step response

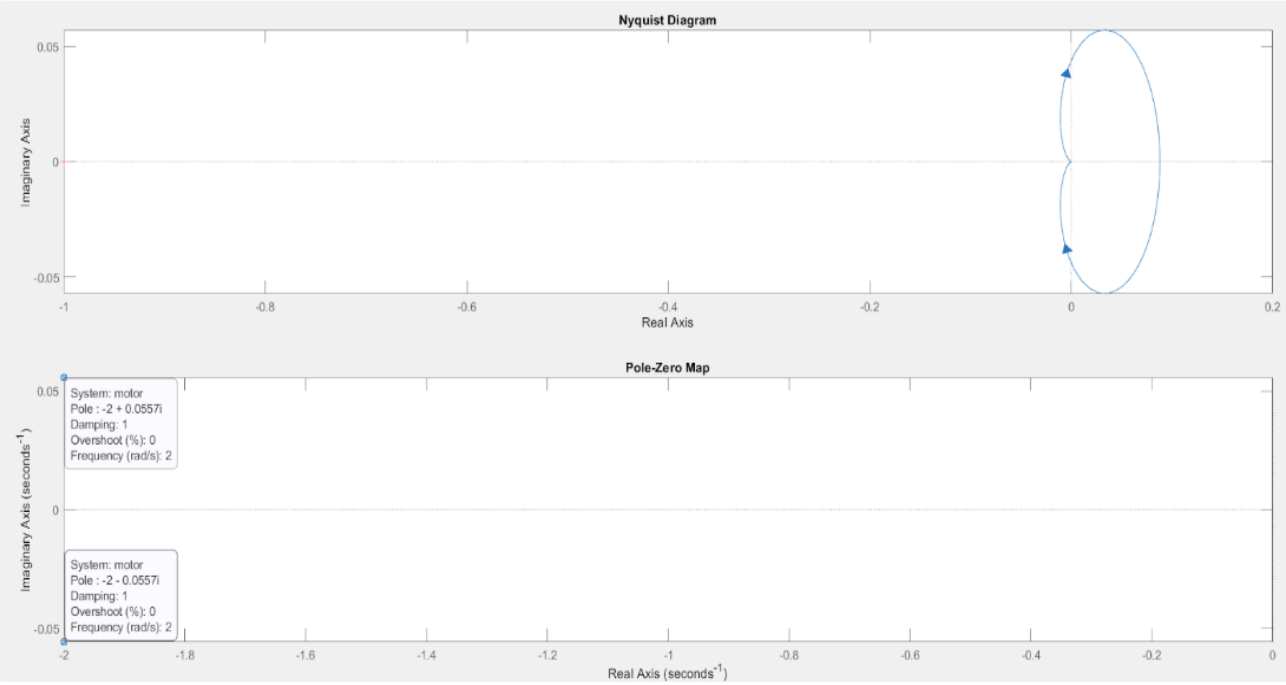


Figure 4.3 Open loop 49yquist diagram and pole zero map.

Figure 4.3 shows the system's output angular velocity over time of the 1V input where the maximum angular velocity is 0.085 radian/seconds. The DC output motor response gradually increased eventually reaching the steady state after around 2.6 seconds. In ideal virtually a good system would need to have a less than 1 second's overshoot for a stable system with first-order reaction.

Figure 4.4 of the Nyquist plot diagrams show the system responds to different input frequencies regarding stability and the resonance of the motor. There is a a small loop in the near real region axis which suggests a system's frequency response to be stable without any oscillatory behavior . The pole-zero map indicates the location of the pole to be -2+/-0.0557i where -2 value is the real part laying I the negative region which confirm the system is table where the imagery part indicates the systems to show very small oscillatory behavior which confirms the finding of Figure 4.3.

In Figure 4.5 a system simulation of dynamic with the Brushless DC motor with a double control signal with feedback and without current feedback. The double signal composed of two domains where a first positive pulse equates to a command to increase the rotational velocity of the wheel and the negative pulse is a command to decrease the rotational velocity. This captures the short time scale around 20 millisecond which is a fast electrical transient. The first row graph presents wheel angular velocity where it responds immediately to the torque input. While the right column with the current feedback has a slower initial acceleration due to the back EMF electrical lag and small overshoot. Fort the electrical state which is the second row the current feedback motor show every high frequency transient meaning it increases then drop back due to control feedback. For the motor torque graph which is the third row. The current feedback shows that the torque does not immediately apply due to electrical inertia. This is very helpful in modeling a precise control system where the systems doesn't want to be extreme overshoot in an extreme critical earth pointing mission critical maneuver.

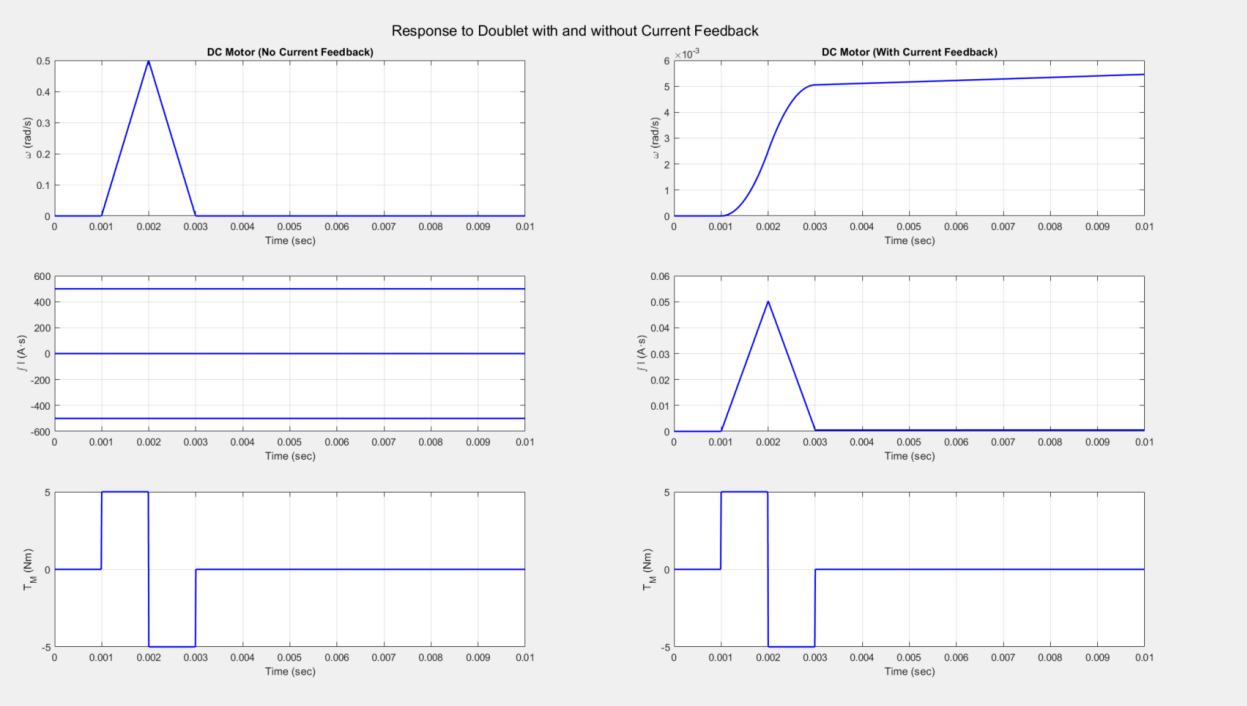


Figure 4.4. Doublet with and without current feedback from current sensor.

4.2 Configuration Modeling

To map the torque distribution on the x,y and z axes of the pyramidal configuration. There must be an established relationship of the 4 wheels respect to the spin axes.

Such distribution of each wheel can be represented by a 1x3 matrix with cosine and sine for magnitude distribution shown the equations 4.10 4.11, 4.12 and 4.13 which represents wheel 12,3 and 4 respectively. The rection wheel contributes opposite torque to its unit vector in the body frame.

$$B_{z1} = \begin{bmatrix} \sin\alpha * \cos\beta_1 \\ \sin\alpha * \cos\beta_1 \\ \cos\alpha \end{bmatrix} \tag{4.10}$$

$$B_{z2} = \begin{bmatrix} \sin\alpha * \cos\beta_2 \\ \sin\alpha * \cos\beta_2 \\ \cos\alpha \end{bmatrix} \tag{4.11}$$

$$B_{z3} = \begin{bmatrix} \sin\alpha * \cos\beta_3 \\ \sin\alpha * \cos\beta_3 \\ \cos\alpha \end{bmatrix} \tag{4.12}$$

(4.13)

$$B_{z4} = \begin{bmatrix} \sin\alpha * \cos\beta_4 \\ \sin\alpha * \cos\beta_4 \\ \cos\alpha \end{bmatrix}$$

$$\beta = [0, 90\ 180, 270] degrees$$

Beta vector is the angle distribution of 1 through 4 in the order and α would be the tilt angle of 54.74 degrees from the XY plane (horizontal). The reason is to maximize the torque generation in three axis so the sum of all nit vectors would need to be symmetric about the z axis. The azimuth angle would be 120 degrees to prevent bas in one direction and maintain the XY plane symmetry.

Using MATLAB the Figure 4.5 can be modeled to be shown of th 4 reaction wheels motor modules to be placed on the spacecraft. This would mean that each of the wheel will have affect on x,y and z axes therefore creating redundancies systems.

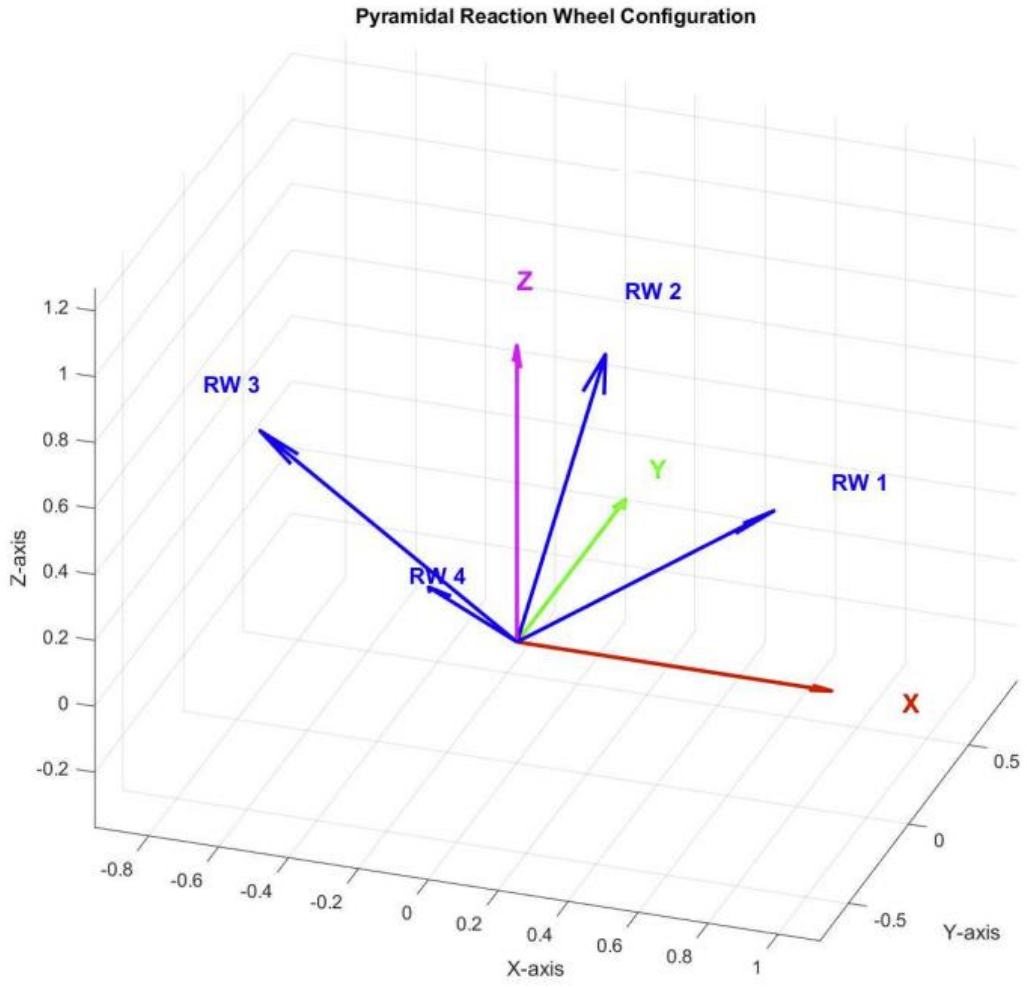


Figure 4.5. MATLAB visualization of the wheel distribution in pyramidal configuration.

The torque on the spacecraft from the 4-reaction wheel can also be modeled in equation 4.14.

$$ {}^{B}\tau = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = - \begin{bmatrix} \sin\alpha & 0 & -\sin\alpha & 0 \\ 0 & \sin\alpha & 0 & -\sin\alpha \\ \cos\alpha & \cos\alpha & \cos\alpha & \cos\alpha \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \\ \tau_4 \end{bmatrix} \tag{4.14} $$

4.3 Three Axes Control Algorithm

Each axis has a PID controller where the systems would reference a desired quaternions input and a measured current quaternion from sensor. The quaternions would then be multiplied to produce a small angles error. This proportional integration differential controller generates a per axis acceleration demand to each of the plant subsystem which is the rection wheel motor. This then is multiplied with the inertia matrix of the body satellite to produce a torque command. If the systems contain thruster, there would be a pulse width command using linear programming for simplex algorithm. However, if there are no thruster the default state would be the reaction wheel command. This would essentially convert to current and voltage feedback into the reaction wheel. Such flow diagram can be shown in Figure 4.6.

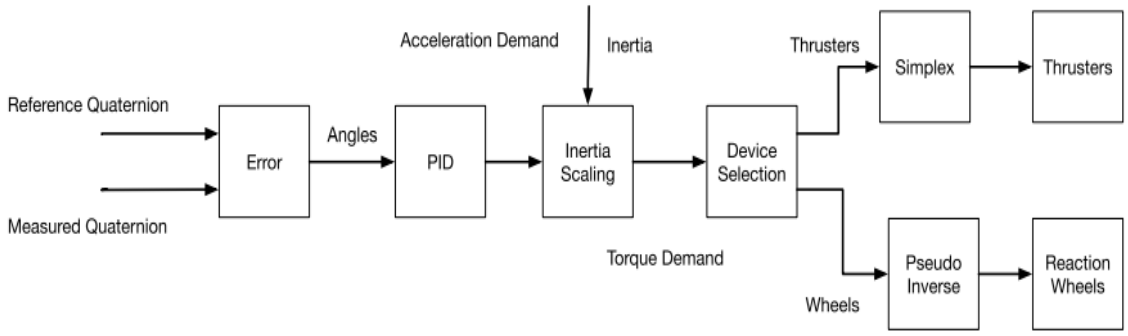


Figure 4.6. Three axes control algorithm.

Without using any PID controller the following graph can be obtained to visualize the tracking vs targeting quaternions of the spacecraft's body state. Figure 4.7 shows that the spacecraft starts with an target 60 degrees rotation about the Z-axis. The quaternion PD controller drives the spacecraft dynamic from initial to target orientation. The reaction wheel generates torque to correct for the quaternion error by using matrix multiplication fo quaternions to obtain error. The quaternions contain 1 scalar and 3 spin components. The q1 plot shows that q1 starts a t a large value but since the target is rotating about Z q1 must decrease toward 0. This indicates the spacecraft is reducing its rotation around X axis as it approaches the Z axis rotation. The q2 represent y axis spin rotation. The systems initially has 0 y axis component so q2 stays close to 0 at all time. Q3 shows increasing in value meaning that the spacecraft is rotating around the Z -axis. Q0 is basically the most important indicator where it can tell the mission control if the desired quaternion is approaching rapidly or not.

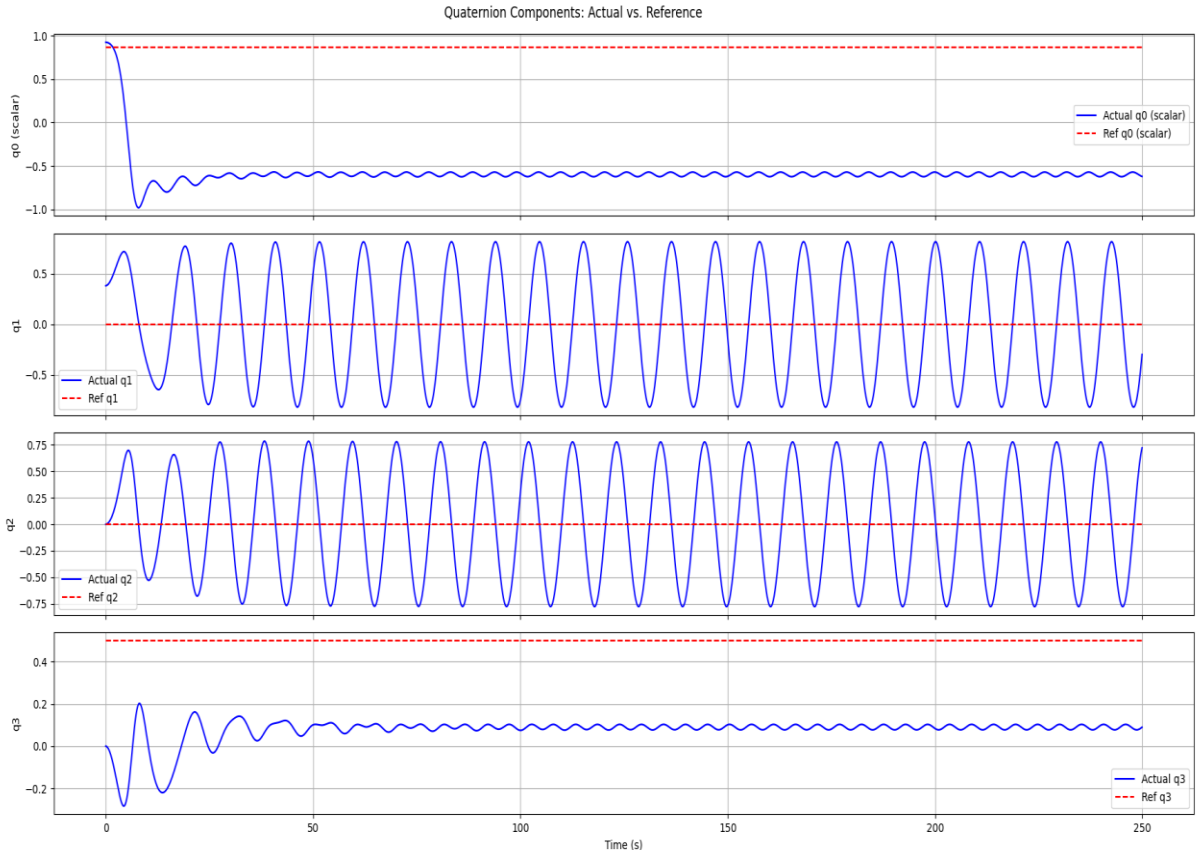


Figure 4.7. PD systems of 45 degrees command around the x- axis.

4.4 Torque and Momentum Control Loop

The torque control loop can be modeled by using a unified modeling language (UML) shown in Figure 4.9. The input of the script uses a simple forward Euler integrator with a time step of 0.1 seconds over a simulation period of 50 seconds. The spacecraft has inertia of diagonal matrix of I = diag[10, 14, 8] where the reaction wheel has a spin axis inertia of 0.06 kg*m^2. Since there are no cross product of the inertia such that there are no diagonal terms within the inertia matrix, the rotational cross-coupling only comes through the reaction wheel and not from the spacecraft's inertia itself.

The reaction wheel configuration and arrangement once become a 3x4 matrix such that there is a no unique solution for any 3-dimentionsal torque demand. A function of Moore-Penrose called pseudo-inverse to find a best fit set of wheel torques.

The control gains function use a simple PID on the attitude error and rate error such that the current Kp = 5 and Kd = 3. This essentially accumulates the term of the quaternion vector error.

For initial condition the is no angular velocity. The spacecraft starts with a 32 degrees rotation about the x-principal axis and a nonangular velocity of w0 = [0.1, -0.05, 0.005]. This would also bring four reactions wheel to at nulling spinning speed.

For the main loop computation, the code ensure that the current quaternion is normalized to avoid the floating drift of a point. The attitude error calculation is applying the matrix multiplication of the desired quaternion with the conjugate of the current quaternion. The PID control then takes over to accumulate the integral of the attitude. Then a torque function is called to produce a quaternion vector error. The wheel torque allocation essentially utilize pseudo inverse and then angular momentum is updated using the Euler's equation. The current sensor is also being used to get the current feedback to determine the exact torque being produced as well as power being drawn from all of the actuators.

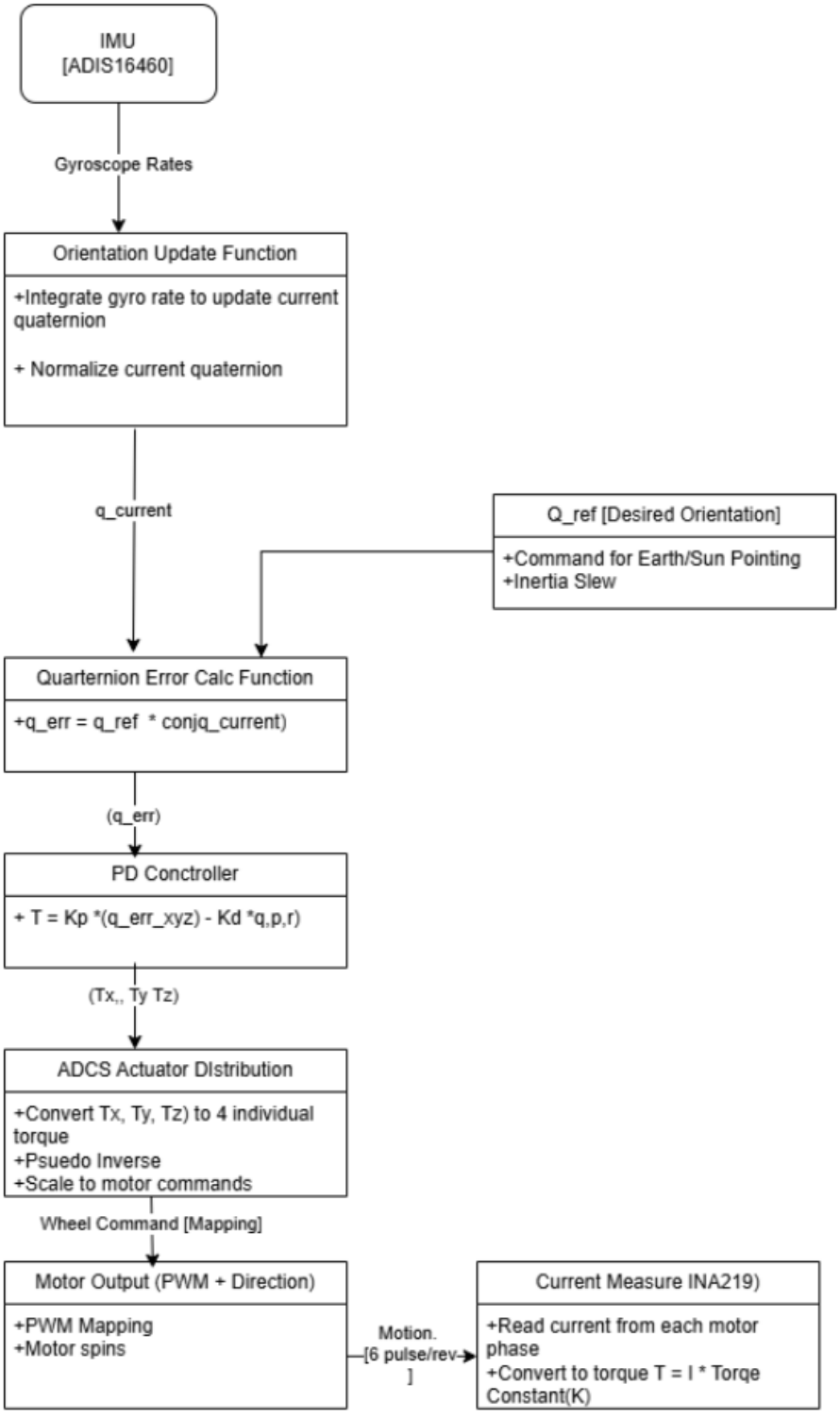


Figure 4.8 UML diagram of torque control code.

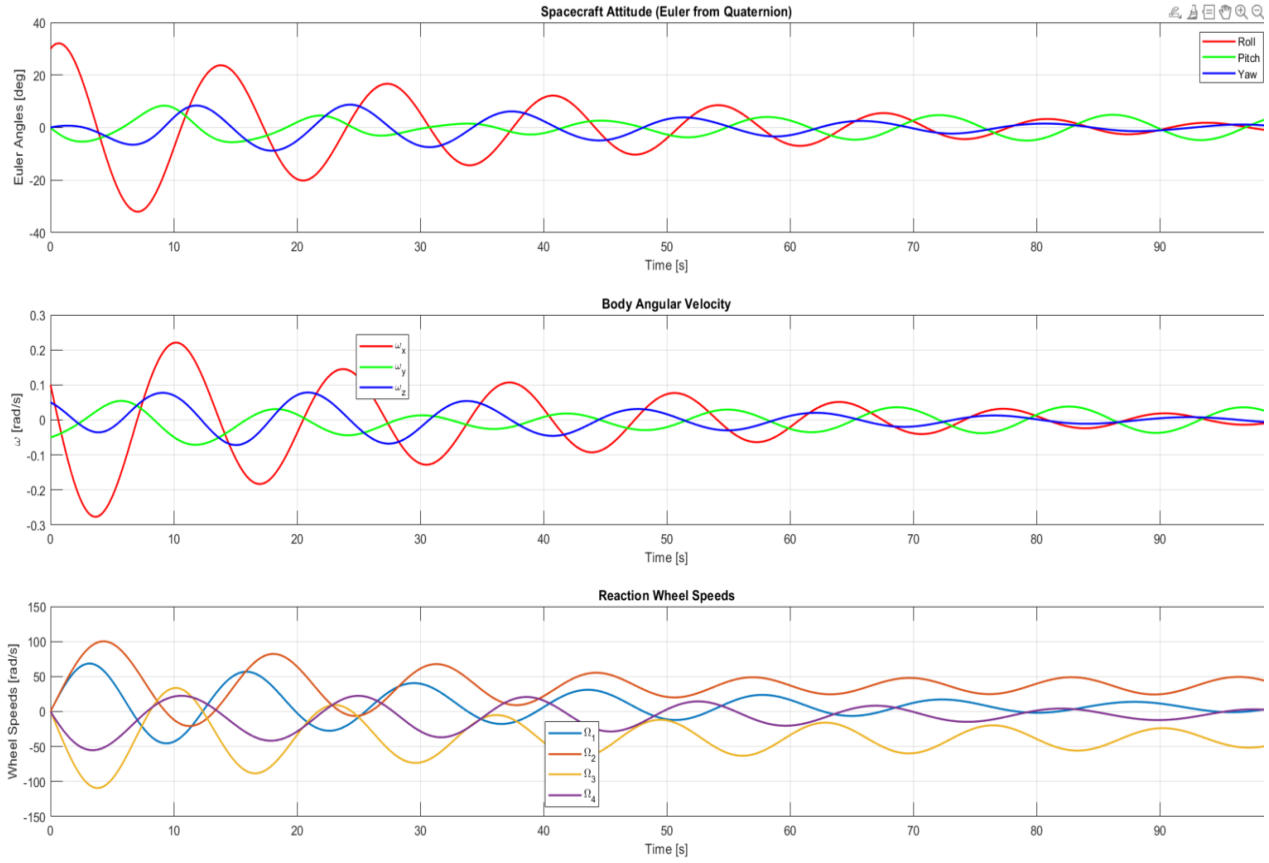


Figure 4.9 Spacecraft simulations based on the control UML code blocks.

In Figure 4.9, the spacecraft is clearly to be at a 30 degrees roll offset with a little perturbation of body-rate offset. As time goes on the PID controller is attempting to drive the system back toward zero Euler angles since there is no external damping of motion as the Kp and Kd increase. There is no energy being dissipated where the system can continue to exhibit sinusoidal oscillations in three axes near 0.

The angular velocity graph shows a oscillations where the gain factor is overshooting which cause the controller to correct in each axis. The red curves show the greatest amplitude where the 30 degrees is initial roll offset. However, there can also be observed cross coupling appearance within pitch and yaw due to the quaternion control law within the tetrahedral wheel configuration.

The reaction wheel speed graph also shows the sinusoidal behavior as the wheel speed up and slow down as the response to the body's rotation. The total angular momentum is also conserved since there are no external torques where the wheels and the spacecraft effectively exchange momentum back and forth. Since there is a positive and negative amplitude there is no saturation that occur within this maneuver.

4.5 B-dot controller for Detumbling

The principle B-dot controller is use to control the detumbling nd reduce angular momentum by generate a magnetic torque from the rate of change of the measured magnetic field of the body frame. The principle can be shown as:

$$M = -\frac{k\frac{db_{body}}{dt}}{||body||} \; and \; T = M \; x \; b_{body} \tag{4.15}$$

A Matlab script is written such that the B-inertial can be representative in micro tesla of 1x4 matrix. The discrete B-dot should be computed as equation 4.16 such that it represents the step size.

$$b_{body} = \frac{b_{body}(t+\Delta t) - b_{body}(t)}{\Delta t} \tag{4.16}$$

The command dipole can be formed by

$$M = \; -gain * \frac{k\frac{db_{body}}{dt}}{||body||} \tag{4.17}$$

The Euler rigid body equation would need to be integrated to update the quaternion after.

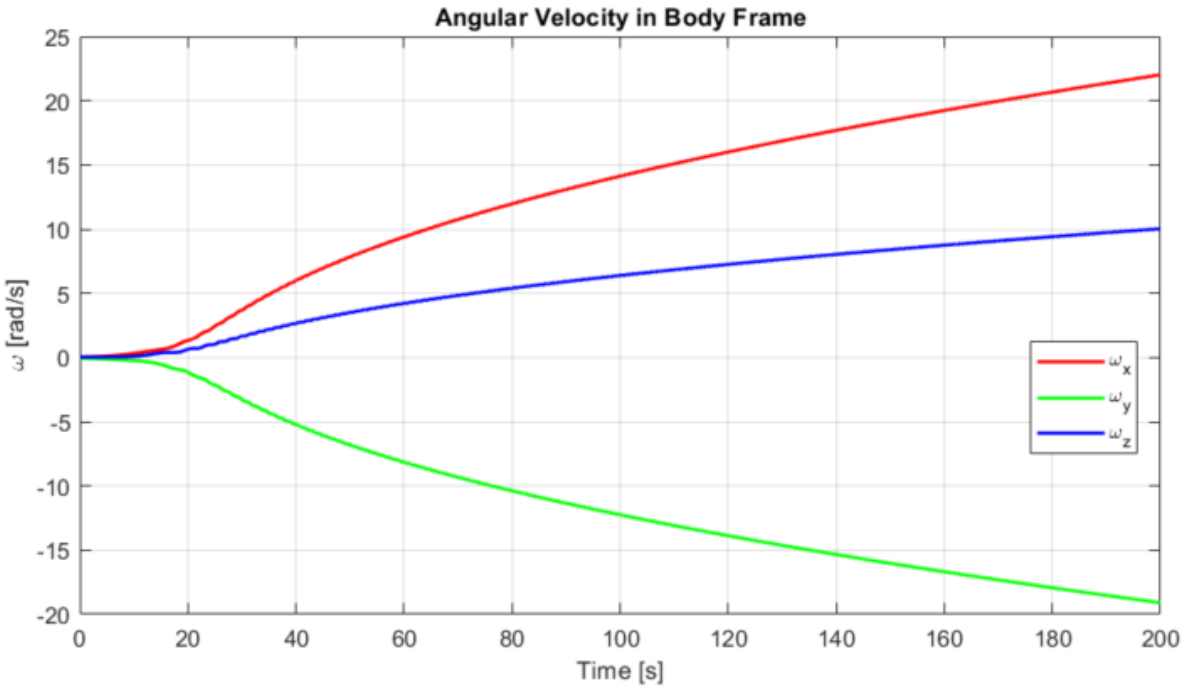


Figure 4.10 Angular velocity in body frame with high gain

In Figure 4.10 an the angular velocity in Body Frame show an ideal B plot in detumbling would show a decay however, there shown to be extreme injection of rotational energy than damping it. This is prove to show that the critical gain factor is super important in these early orbit operation. This also indicate eh tenet roque has a sign error or wrong polarity within the hardware. There can also be an unrealistic assumption such that causing a large gain which oversaturates the system.

The magnetic field plot in Figure 4.12 and 4.13 show the variance from 3x10^-5 Tesla to -3 x 10^-5 Tesla in the Bz axis where it is a quick spinning axis. As the satellite speeds up the body changes more rapidly so the derivative will become larger as the amplitude shown. The command dipole also showing a ramp up in amplitude such that the magnetorquer rods will have a strict current and voltage limit to prevent such large dipole command.

A possible solution for this problem will be to simplify the Eci and use small angle approximation as well as ensuring the M x b calculation is an opposition rotation for the sign convention. There can also be some damping being used such as internal friction to be modeled which can always subtract energy from the body.

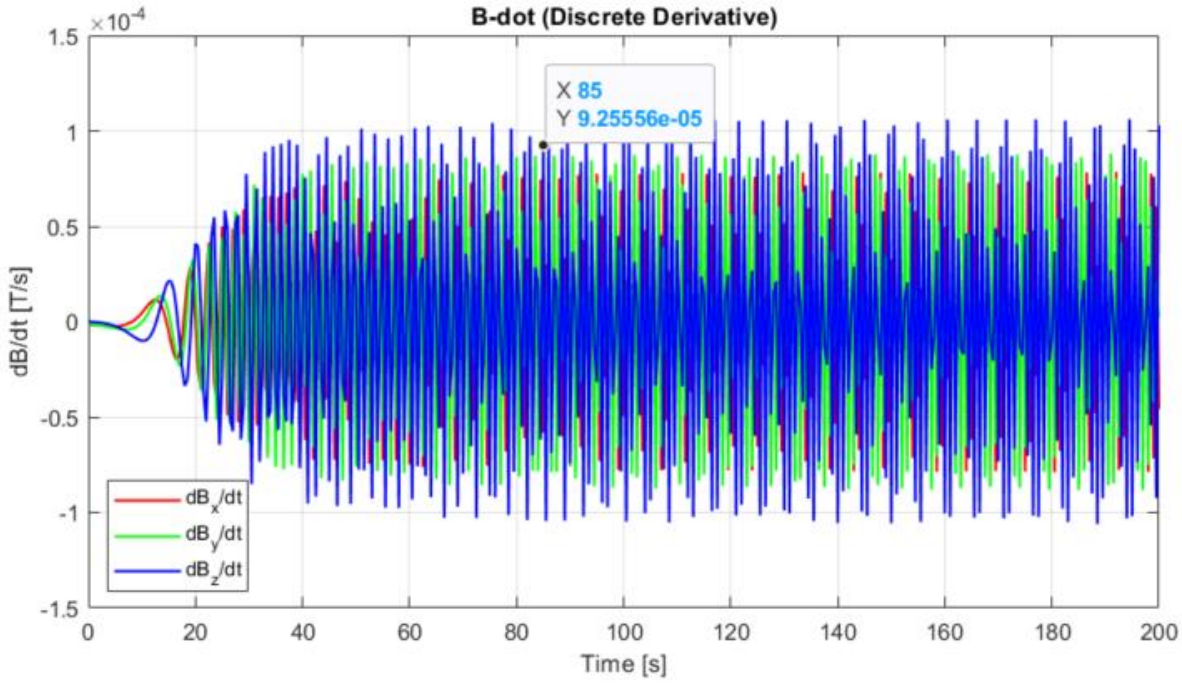


Figure 4.11 B-dot derivative.

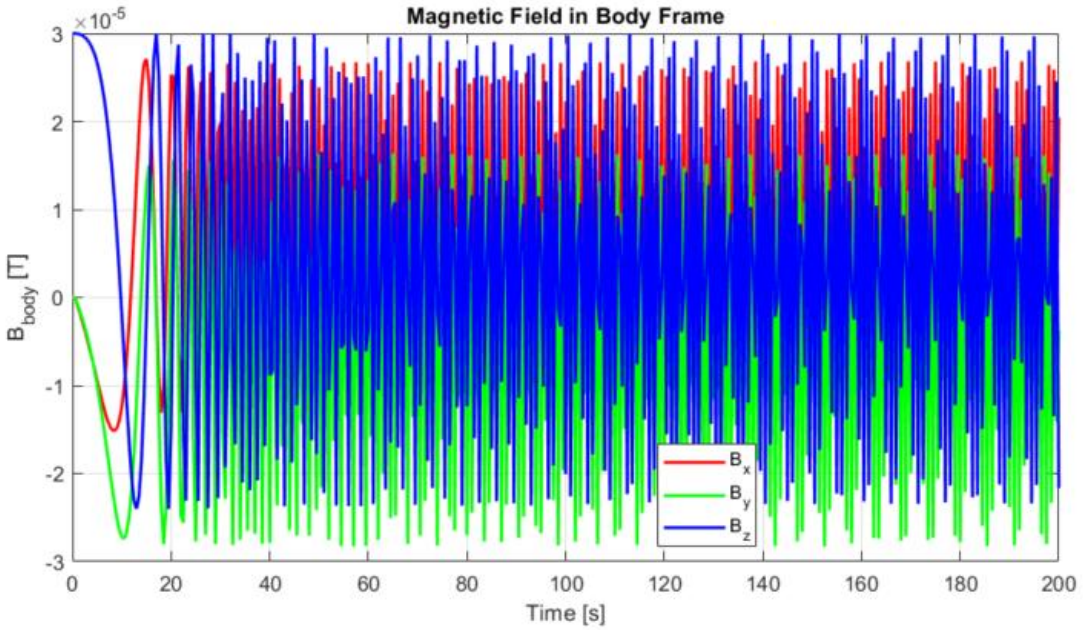


Figure 4.12 Magnetic field in body frame.

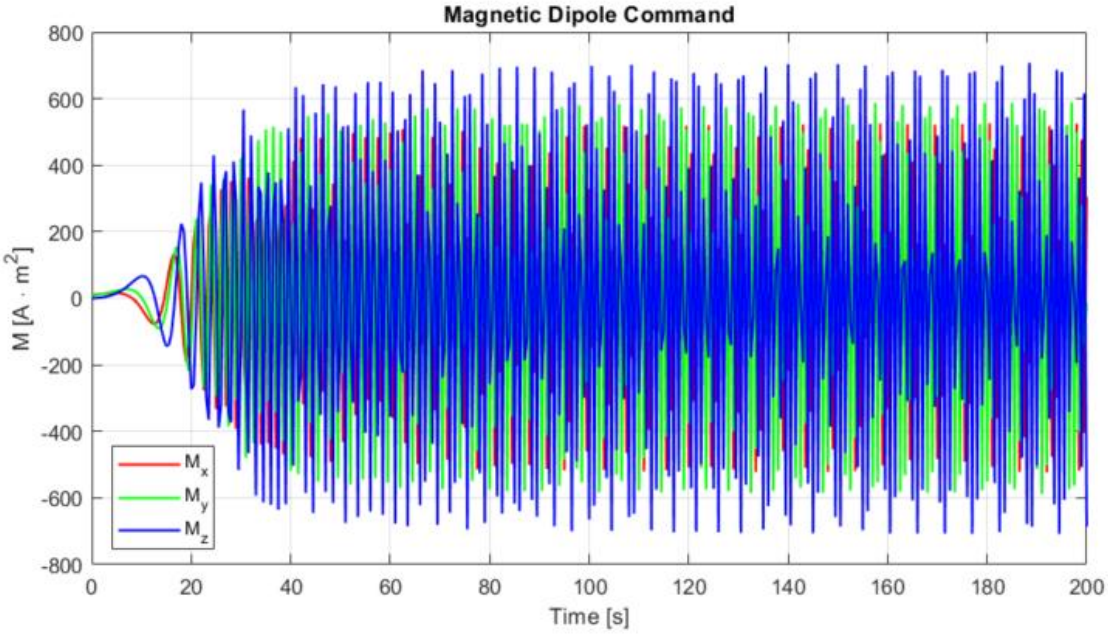


Figure 4.13 Magnetic dipole command.

5. Mission Mode Simulation

The attitude control mode so used to control the various GNC process and transitions during the life of the TechEdSat spacecraft's mission. The attitude control mode is a configuration of the GNC bus flight software subsystem for a particular mission phase. The following table can be used to describe such modes.

Table 5.1 GNC Mode

Mode Name	Sub-Mode	Description
Standby		The state of the vehicle at standby mode allow for flight software initialization. 1. The spacecraft onboard computer as well as ground operator has the ability to disable attitude control in which the disabling momentum adjust, initialize the torque filtering, zero torque command and attitude error and rate error. 2. Spin up and perform wheel self-test enabling current and proper wheel ID being tested 3. Spin up test 4. Open the options for the ascent deployment and recovery contingency
Detumbling		The detumbling mode objective is to fully autonomously reduce and dampen the satellite's angular velocity. Using the b-dot algorithm from the magnetorquer of the iMTQ board to fully established.
Inertial Slew		After the adjustment of the detumbling the spacecraft shall align on a specific body axis nominally the Z axis which is nadir. The target vector shall rotate at the commanded rate to be 0.5 rad/sec. The spacecraft then shall rotate to maintain at the rate over the inertial frame.
Nominal		The nominal control mode consists of the torque and scientific quaternion input to drive a specific spacecraft to point. The attitude control drives the vehicle to closely follow the commanded quaternion. This will also control solar array to track the sun. This will also support the mission operation orbit management and momentum control rate.

	Calibration [Built in Self-Test]	The sub-nominal mode of calibration with built in self-test for every single sensor and actuator on board will carry out set of basic test to check the telemetry and health before enabling command for slew. Gyroscope will have certain electrostatic force being applied to each of the spin axis to enable the readings. The magnetometer will have x-y and Z sensor magnetic field be calibrated before the sensor fusion.
	Scientific Pointing	The scientific mode will take in a very specific quaternion commanded input to plug into the PID loop from the torque control loop to use the 4 ADCS reaction wheel to point to a very specific target.
Sun Acquisition		The sun acquisition mode is enabled to maintain the sun safe attitude and rates with the digital sun sensor array and reaction wheel assembly. The spacecraft shall not be stowed which means the solar array must always point at the sun. The deployed sun acquisition align to the body target which also slew the solar array to sun by using the sun finding mode.
	Sun Finding	The sun defined mode execute a rotation to search the sequence of the body Y axis rate of 2 rad/sec to locate the sun
	Sun Maneuver	Once the sun is found the spacecraft shall slew down to sun safe that is dependent on the solar array deployment state matrix.
	Sun Wait	If the spacecraft and the course digital array cannot find the sun, then loop back in the command to find it again .
Safe Mode		During safe mode command and detect the failure. The OBC has the option to enter recovery and configure the GNC mode to enter sun acquisition where the solar array are fully deployed and sync with the spacecraft. There is also a stowed safe mode to tug in the solar array for further safety. Once the solar array is tugged in there will not be any slew rate commanding enable.

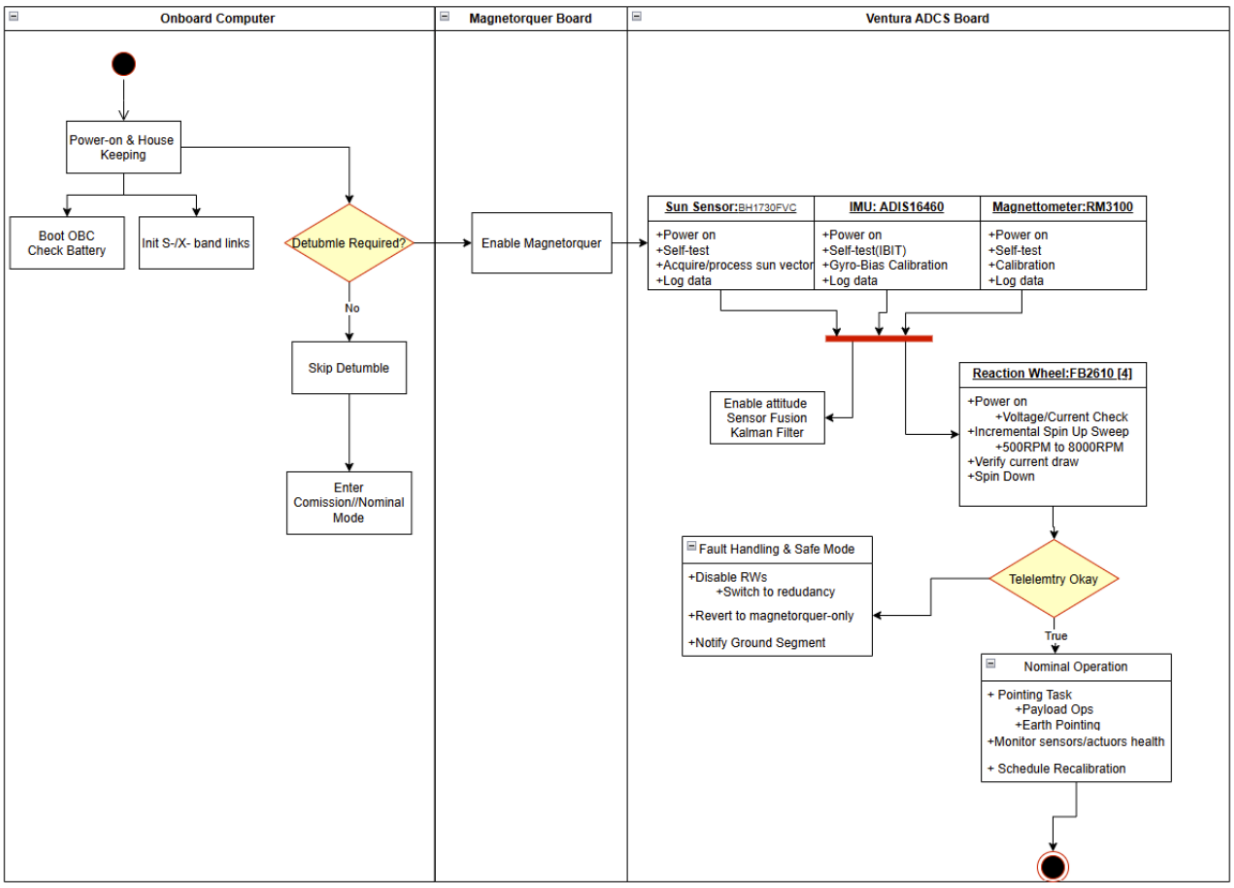


Figure 5.1. State machine diagram of ADCS operation.

The state transition starts off with the initialization of the bus to enable communications with the OBC to switch to detumbling mode after spacecraft separation. The detumbling mode uses the B-dot controller to stabilize the angular velocity of the spin axes. The OBC then tell the ADCS board out switch to the nominal operation of the spacecraft such that self-tests and self-calibration occur for all sensors and electromechanical on board. If the sensor fails to meet the passing self-test criteria, the ADCS board will run diagnostic check and go back to standby mode for 1 minute. If it cannot get out of safe mode, the board will go into safe mode until the onboard data handling unit response to the recovery operation. The recovery operation includes all of the safe mode stowed as well as sun search operation.

However, if the diagnostic list is checkout the nominal trigger is activated and the ADCS board will enter nominal mode. The nominal mode can now be simply receiving a quaternions command to position the spacecraft to a direction using the PID torque control loop.

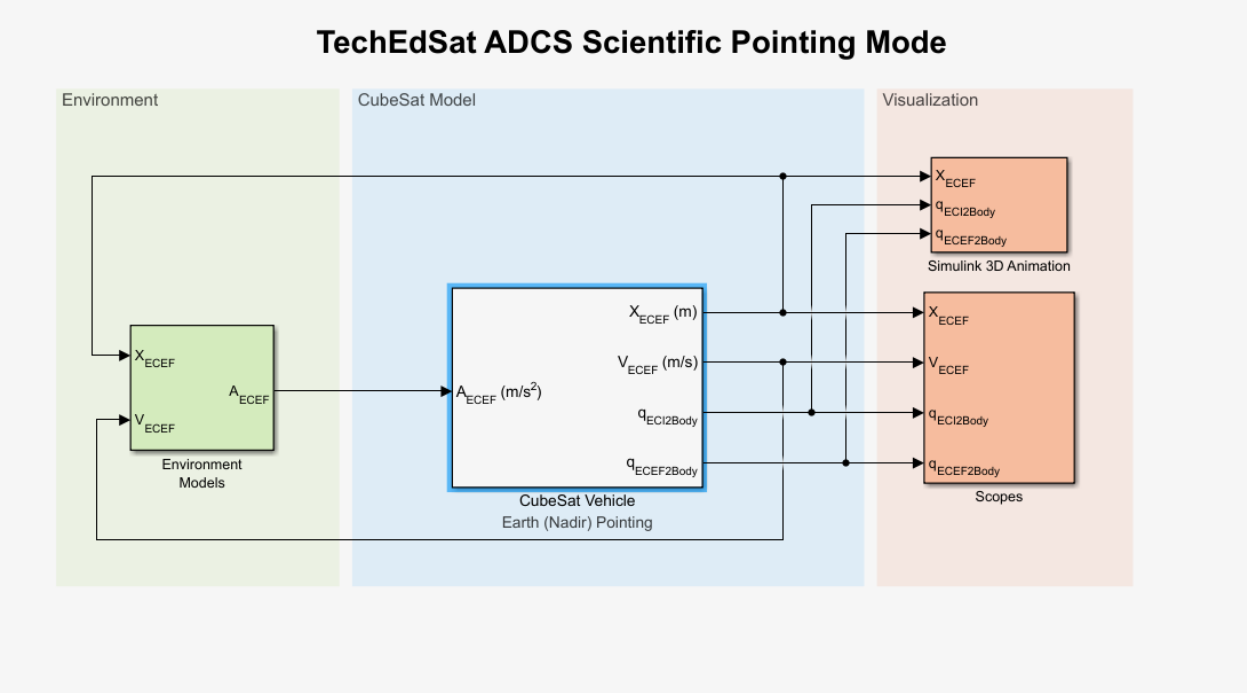


Figure 5.2 TechEdsat ADCS nadir pointing Mode [31]

This Simulink model is generating eternal output that can be described in a table 5.2 below:

Table 5.2. Parameters Definition

Parameter	**Reference/Frame**	**Description**
Position (ECEF)	Earth-Centered Earth-Fixed (ECEF)	The satellite's position relative to Earth's center, which is expressed in a frame that rotates with Earth. This is sed for ground-track calculations and helping with allocating and determining the local vertical (nadir) direction for pointing.
Velocity (ECEF)	Earth-Centered Earth-Fixed (ECEF)	The satellite's velocity to be respected in the rotating ECEF frames. This is crucial for orbital propagation and many attitude-control algorithms.
Acceleration (ECEF)	Earth-Centered Earth-Fixed (ECEF)	This is Acceleration due to gravity and other in ECEF coordinates which is to be used to integrate the orbit or simulate onboard sensors (like an IMU, Magnetometer).
Quaternion (ECI→Body)	Earth-Centered Inertial (ECI) → Body	The spacecraft's orientation with respect to an inertial Earth-centered frame. Useful for inertial pointing (e.g., star trackers or sun sensors) and general 3D attitude knowledge.

Parameter	Reference/Frame	Description
Quaternion (ECEF→Body)	Earth-Fixed → Body	The spacecraft's orientation relative to Earth's rotating frame. Critical for nadir-pointing, as it defines how the local vertical maps into the body axes.
"Earth (Nadir) Pointing"	ADCS Mode Setting	An operational mode in which the CubeSat keeps a specific face (e.g., +Z axis) aligned with Earth's center. Uses ECEF data to track the nadir direction.
Environment Model Outputs	ECEF (and possibly others)	Earth orientation parameters, and other orbital dynamics. The CubeSat block consumes these to propagate its orbit and attitude.
Visualization Inputs	ECEF/Body/Quaternion Data	The 3D animation and scope blocks receive the updated states (position, velocity, quaternions) to display real-time orbit and attitude plots.

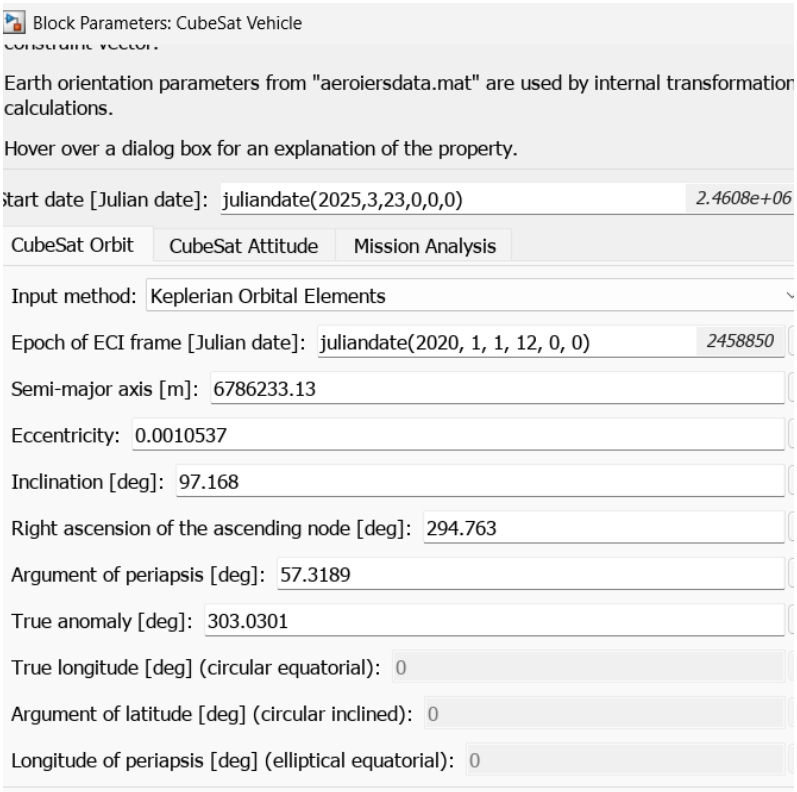


Figure 5.3 Orbit parameter tab

Notable parameters would be the Keplerian orbit elements which is defined by classical parameters such as semi major axis, eccentricity, and inclination. The semi major axis would be

6786km where this is in the low earth orbit regime. The eccentricity is 0.0010 which is a nearly circular orbit for TechEdsat. The inclination would be 977 degrees which is a near polar or sun-synchronous orbit. The Right ascension of the ascending node is 294 degrees and Argument of Periapsis is 57.3 degrees; True anomaly would be 303.03 degrees such that it defines the orbit's orientation in space and satellite's potion within that orbit.

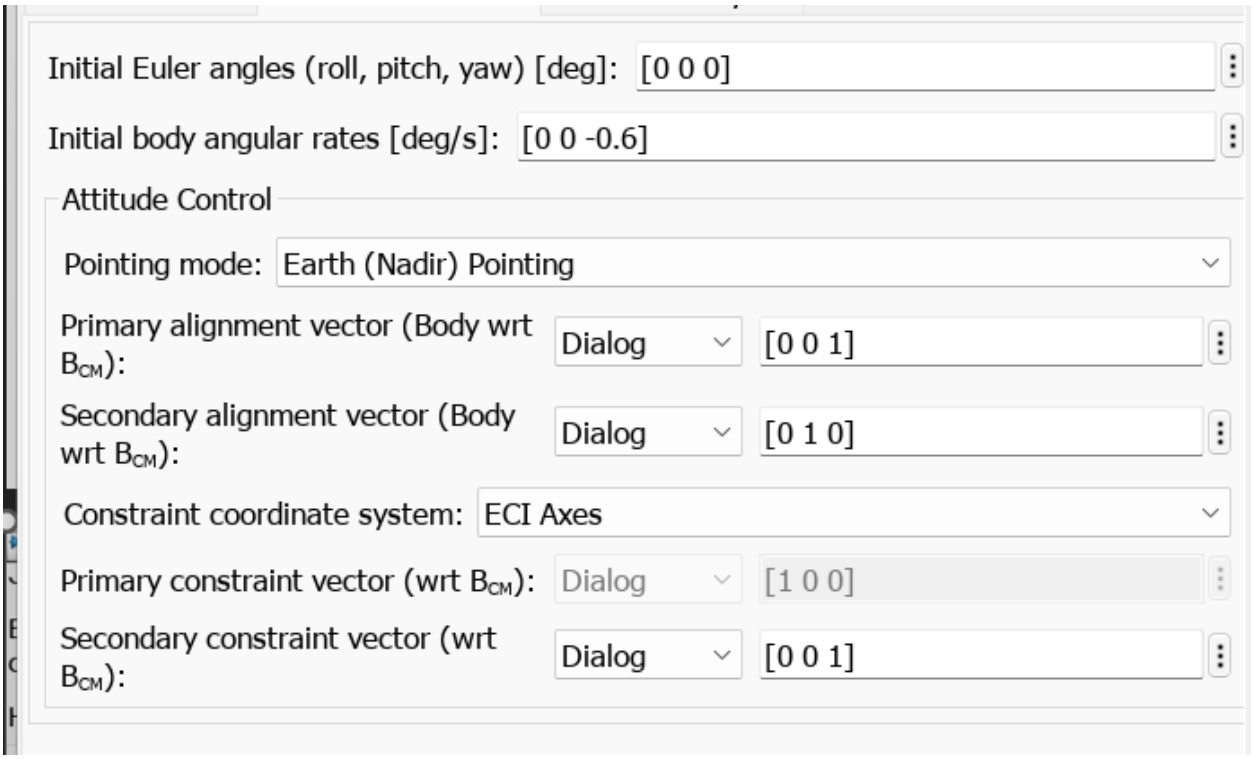


Figure 5.4. Control input parameter

Table 5.3. Orbital parameters

Parameter	**Explanation**	**Importance for CubeSat Orbit/Attitude**
Start Date (Julian Date)	This essential defines the reference start date for simulation, here set as March 23, 2025.	This is essential to synchronize the simulation with real-world time.
Epoch of ECI Frame	Defines the reference frame epoch for orbital calculations (January 1, 2020, at 12:00 UTC).	Establishes initial conditions for the spacecraft's orbit model relative to Earth-Centered Inertial (ECI) coordinates.

Semi-major Axis (m)	Average distance from the spacecraft to Earth center (which is approximately 6786 km).	Determines orbital period and energy of the CubeSat's orbit.
Eccentricity	This is the orbit shape indicator, near-zero means near-circular orbit.	Influences orbital velocity and altitude variations during an orbit.
Inclination (deg)	The angle of orbit plane with Earth's equatorial plane (97.168°), where it shows the sun synchronous orbit	Crucial for mission coverage, Earth observation, and solar exposure as well as pointing to celestial
RAAN (deg)	Right Ascension of Ascending Node, (294.763°), describes orientation of the orbital plane relative to celestial coordinates.	This defines the total orbital plane positioning for coverage planning and sunlight incidence.
Argument of Periapsis (deg)	This specifies the overall orientation of orbit's closest approach point of (57.3189°).	This is important for mission timing related to orbit points where remote sensing is critical
True Anomaly (deg)	Initial angular position of (303.0301°) along orbit at epoch frame.	This is to determine initial location of CubeSat on its orbit path respect to a full cycle at simulation start.
Initial Euler Angles (deg)	Initial attitude orientation (roll, pitch, yaw), set to [0 0 0]. For stable and passive	Provides starting spacecraft orientation for attitude simulation. This would later be used for DCM
Initial Body Angular Rates (deg/s)	Initial rotational velocity of the spacecraft, given as [0 0 -0.6].	Establishes initial rotational dynamics of spacecraft in simulation.
Pointing Mode (Earth Nadir)	Attitude control targeting Earth's center (Nadir).	Defines spacecraft's operational pointing orientation towards Earth.
Alignment and Constraint Vectors	This define spacecraft orientation references relative to body axes and inertial frames.	Crucial for ensuring correct spacecraft pointing/tracking and orientation with respect to Earth and Sun frame.

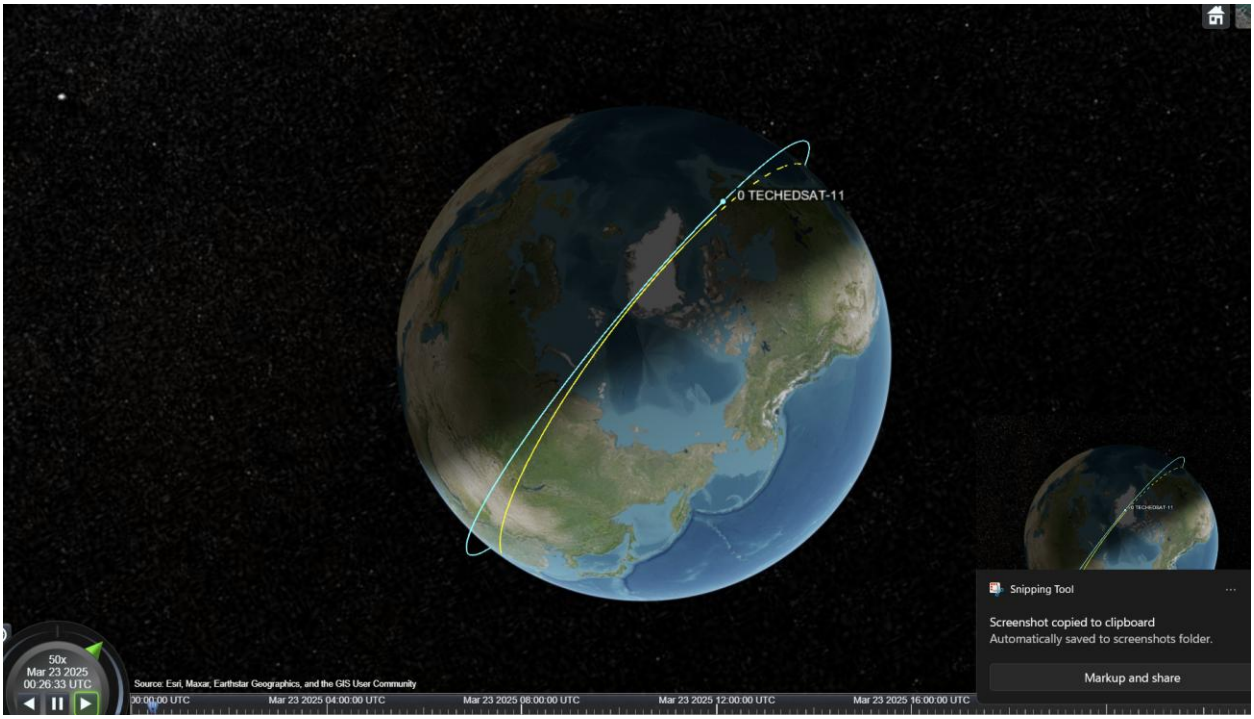


Figure 5.5 TechEdsat in MATLAB STK orbit simulation

Figure 5.5 shows the satellite orbit track, which is the blue line. The yellow line would be the orbit path which it will take per the burn/slew maneuver.

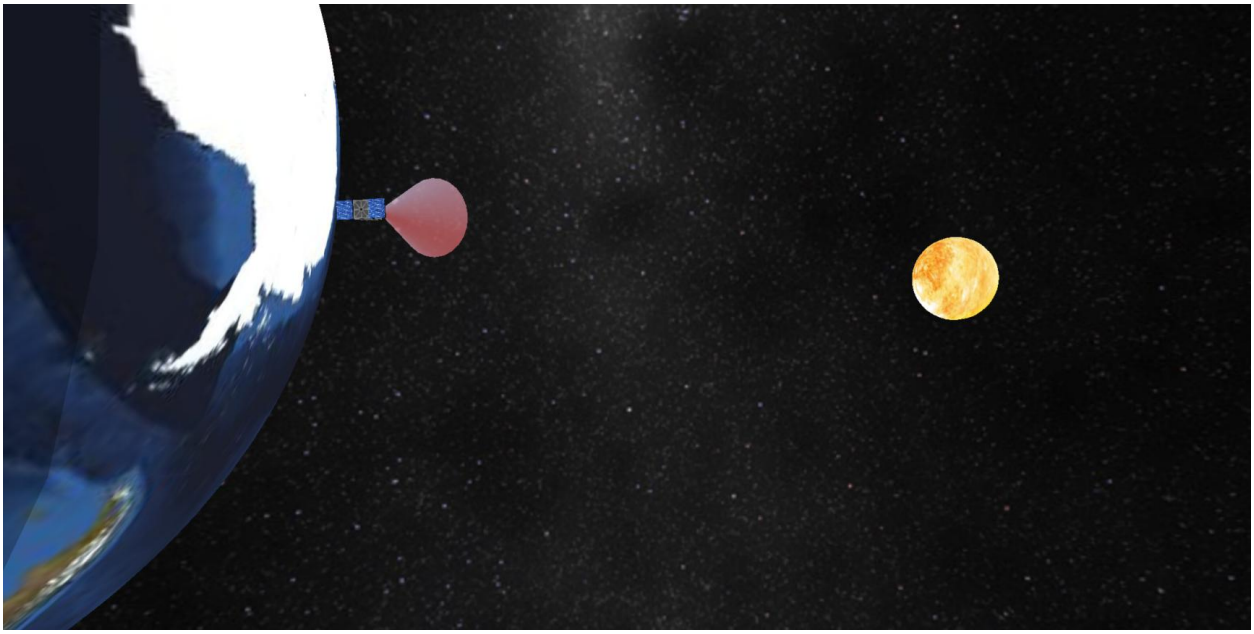


Figure 5.6 TechEdsat sun pointing mode

Figure 5.6 shows the sun pointing mode where the satellite needs to be pointed to the sun for battery re-conditioning. This would also show the sun sensor cone shaped region to be extended from the side. This is directly to gather the sunlight intensity as well as re-calibrate for every 6 months.

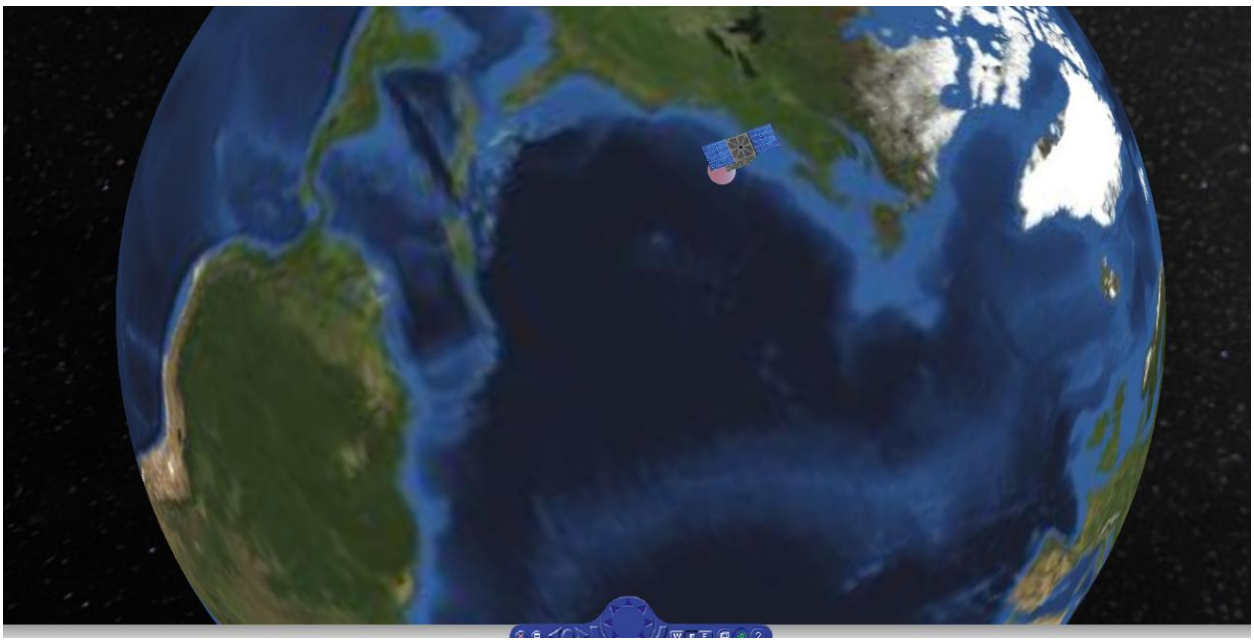


Figure 5.7 TechEdsat nadir pointing mode

This describes the Nadir Pointing Mode such that it means straight down to the earth from the positive Z axis for kept aligned with the line of sight from the ground station. The purpose is to provide earth observation and communication instruments as well as cameras. The satellite calculates the local vertical vector of Xecef and transforms it into body coordinate then an attitude controller will command the spacecraft actuators which are the reaction wheel to vertical align with the body axis.

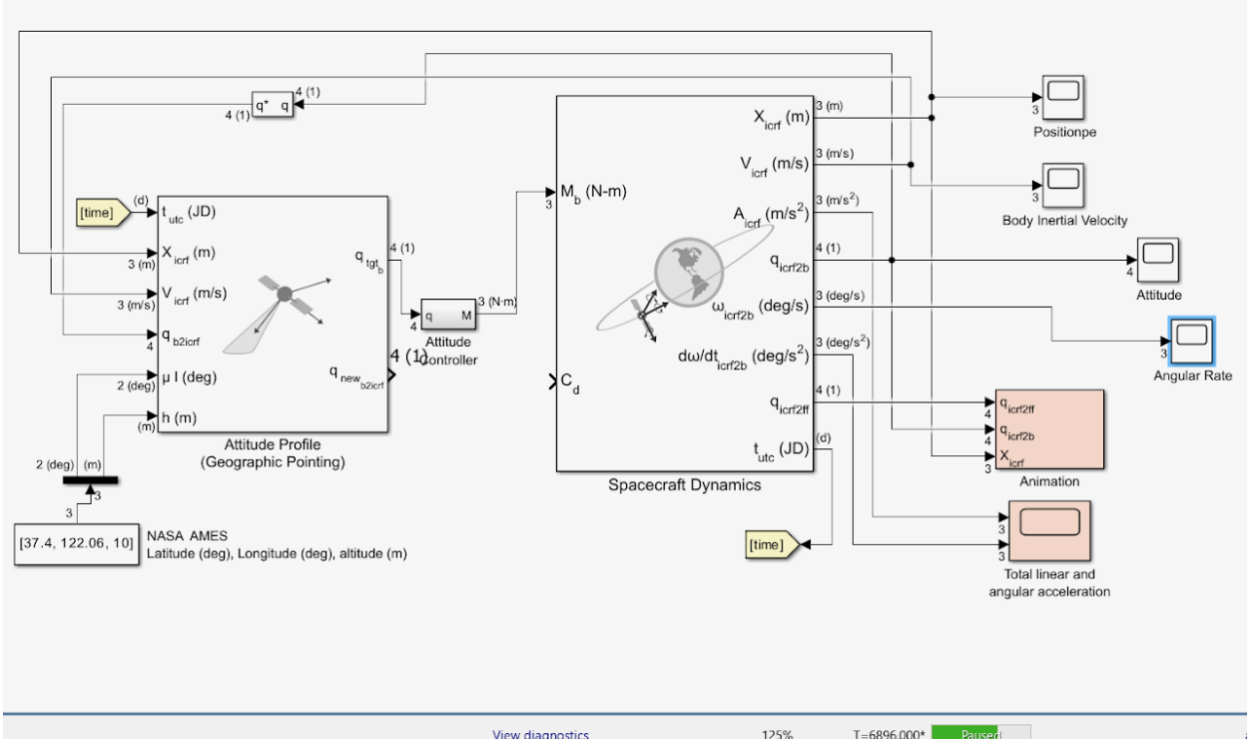


Figure 5.8 TechEdSat with custom coordinate pointing mode

Figure 5.8 shows the Simulink model where a certain coordinate being sent in as a target geography and the block essential contains quaternion conversions to produce torque commands for the spacecraft dynamics. The attitude controller is used and within that is a PD controller. The spacecraft dynamic block receives the control torque which moves around here principally axes Mb in the body frame. External torque is not applied as they are not significantly impacting the spacecraft in nadir pointing. The spacecraft dynamic block also propagates the satellite's translation state and rotational state. This would also output spacecraft body position, velocity linear and angular acceleration for visualization.

The quaternion q_icrf2ff , q_icrf2b and X_icrf are the inputs to animate the orbit state of the satellite. Such that the q_icrf2ff is the orientation of the local vertical and local horizontal LVLH frame relative to the inertial frame. This essential is just used for animation background reference frame. The q_icrf2b is a 4x1 quaternions where is the orientation of the body frame with respect to the inertial frame. This is the spacecraft's current attitude. The Vector X_icrf 3x1 matrix Is the position of the spacecraft I the Earth-centered inertial coordinates in 3D space.

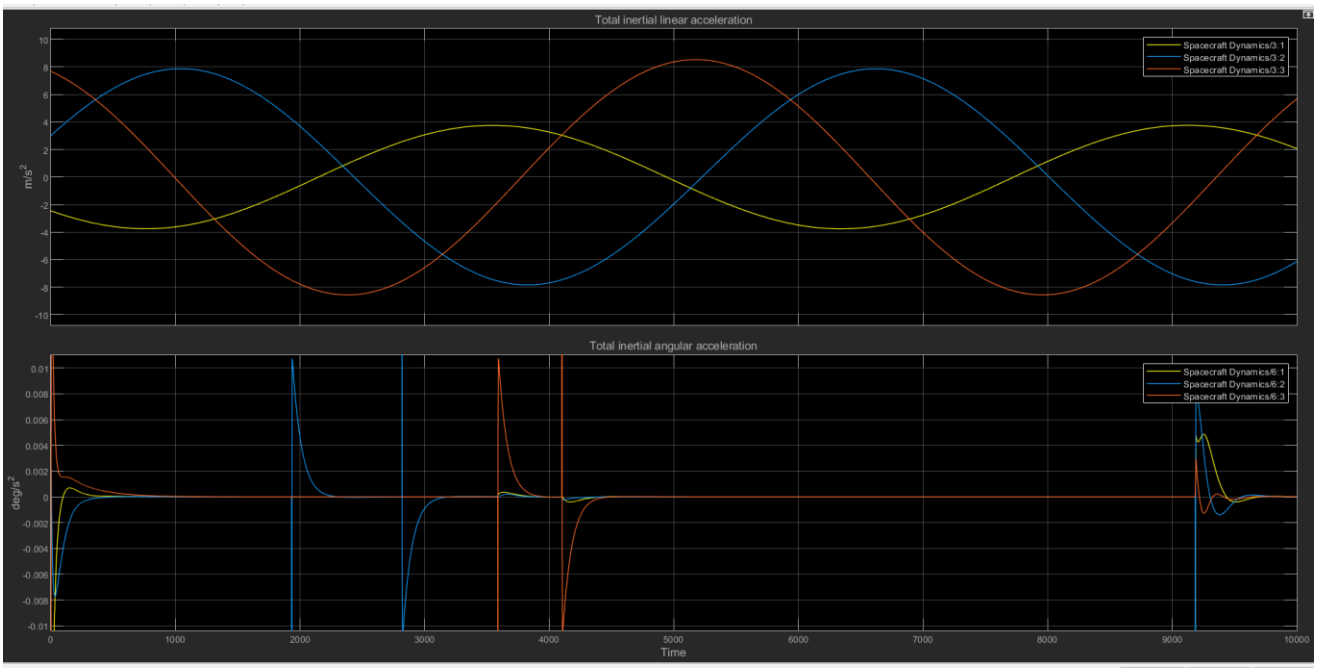


Figure 5.9 Slew rate maneuver

Table 5.4 State timeline

Event Time	**Behavior Observed**
~0–1000 s	Initial condition stabilization right after tumbling
~1900 s	First slew — visible angular spike (blue axis dominant) such that pointing at nadir.
~2900 s	Second slew — possibly targeting 2nd ground station in fast track mode.
~4000 s	Third significant attitude change (orange spike) indicate a fast transient sun tracking mode
~9000 s	Final and largest slew maneuver which has all axes involved; exhibits overshoot and damping due to the back to nadir pointing mode

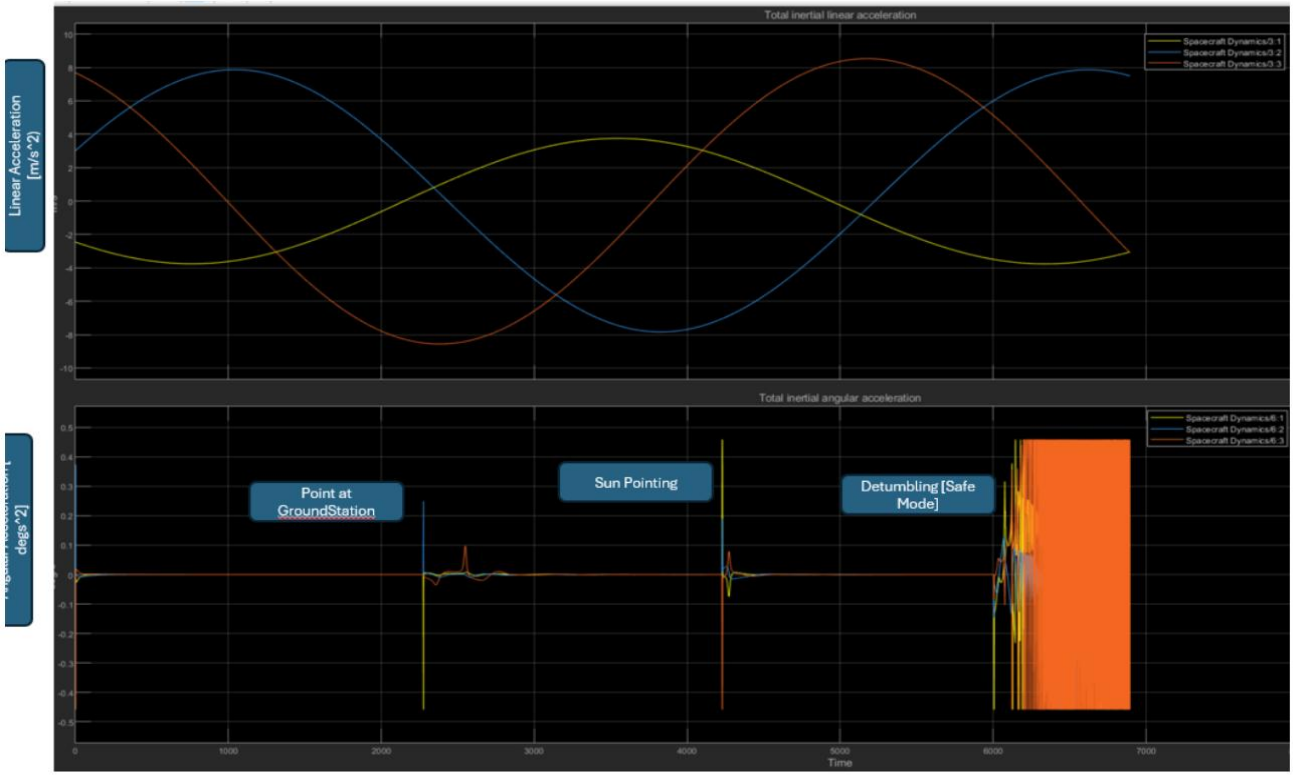


Figure 5.10. Live command sequence

Figure 5.10 is showing the live commanding sequence of the spacecraft. The first command sequence is at point to ground station after detumbling such that it points at NASA AMES ground station. There can be observed of body rate changing in the x, y, and z axis sch that the orange Z axis and blue z axis show positive angular velocity indicating ECEF frame.

The second sun pointing after 4400 seconds causing spikes in all axes especially the x axis as it now is pointing toward the earth. The velocity eventually dampens out over 26 seconds for a smooth operation that does not introduce shock to any component.

The third maneuver is to attempt to deorbit the spacecraft for end-of-life simulation where it spins out of control. This would mean that safe mode is automatically being entered and sending a command down to ground segment to ask to recover or not.

6. Hardware in the Loop

6.1 Testbed Setup

The testbed setup is composed of two major parts which are the Simulink dynamics model and the electrical ground support testbed. The unit under test can be shown in figure 1 which is the Ventura ADCS board. The power distributed to the board will be via the power supply as well as hooking up to a debugging oscilloscope to capture the waveform of the SPI/I2C signal communication. The ADCS module sit on a 6 axis gimbal control bed which can be shown in Figure 6.2 and Figure 6.5.

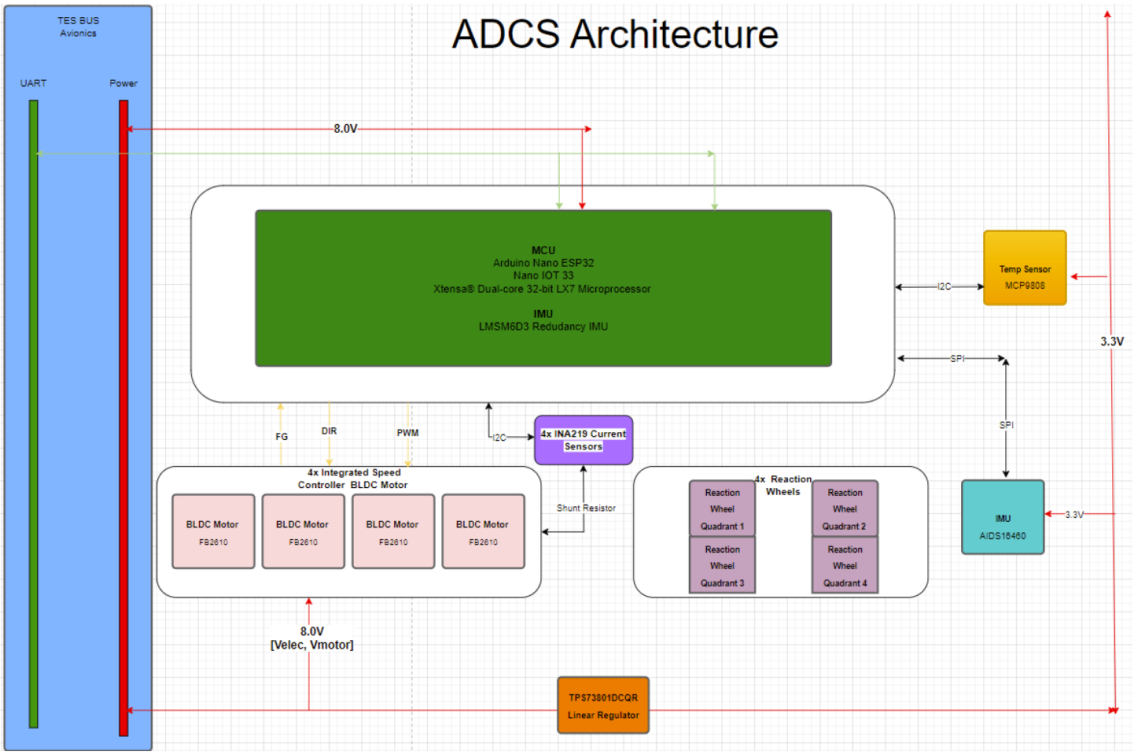


Figure 6.1. ADCS electrical plant interconnection

In Figure 6.2 the testbed setup show an electrical ground support equipment providing power as well as debugging interface to the on-board electronics. The flight computer is being treated as the model on Simulink such that the Ventura ADCS is the plant that is being controlled. The simulant essentially provides the power distribution model dynamics model attitude control signal model IMU model and the 6 DoF model. The controllers receive the model to create registers command passing at 46 Mhz clock rate. Despite not using real time operating systems, the logic loop remained a stacked queue command. The 6 axis gimbal controller for azimuth and elevation will be an independently have its own power supply; however, will share the feedback for verification into the Ventura's IMU model.

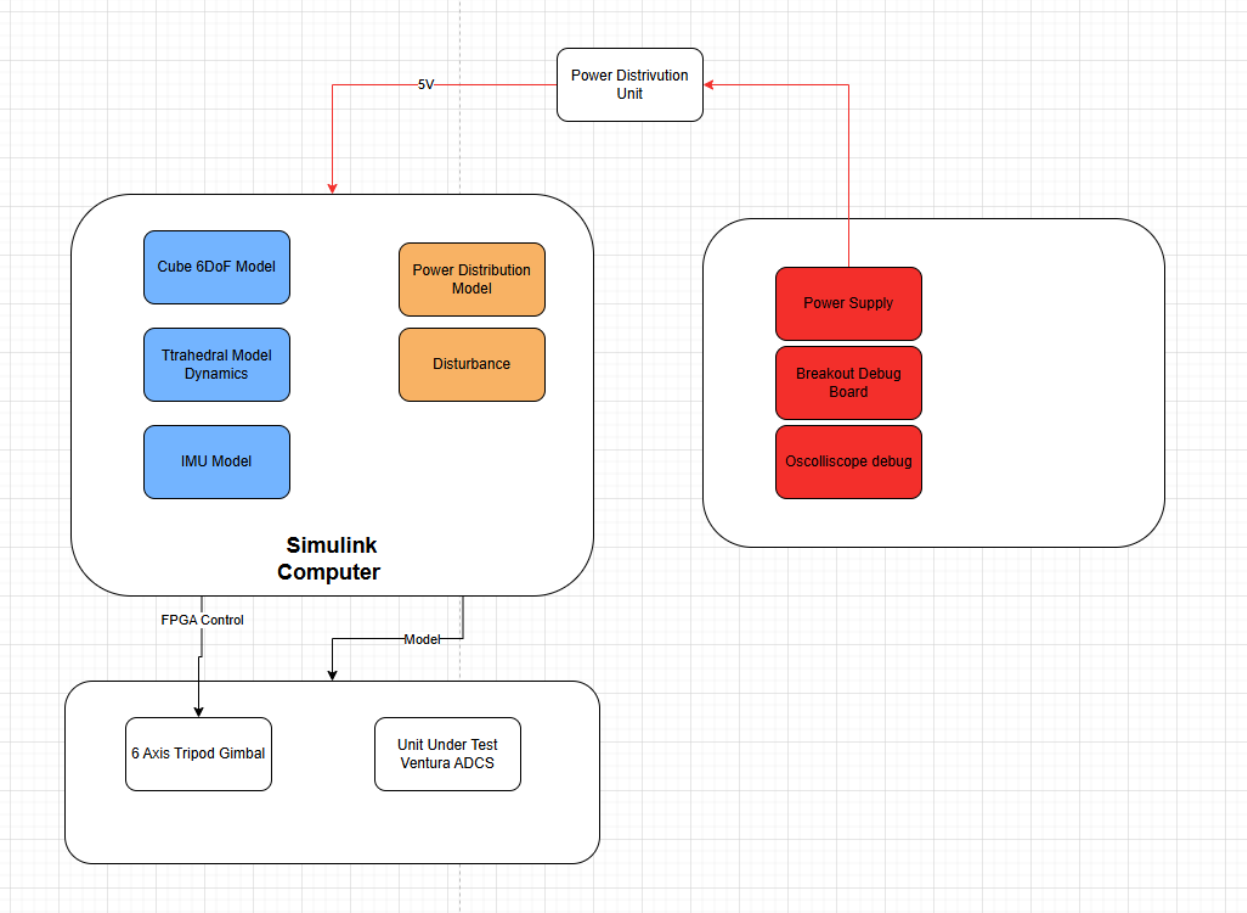


Figure 6.2 Testbed setup

Figure 6.3 shows the unit testing of the sun sensor such that it collects the luminance of the sun sensor in the infrared and visible light spectrum. The two magnitudes were combined to perform self-test as well as entering the control loop. The graph below demonstrates a unit test of 85 seconds of light exposure. The initial phase around 45 to 55 seconds, the lux increases rapidly which indicates a high fluctuation and the turn on command of the sun sensor. The mid phase indicates a drop in lux which is due by the algorithm image filtering and simulating the sun moving away if the satellite is over in safe mode. The final phase is the re-exposure phase to simulate the sudden increase with more sharp peaks followed by another stabilization mode at a 250 lux nominal mode.

Each of the peak corresponds to a exposure event where the sensor are being commanded to rotate toward the light source. The decreasing slope is a simulated path to see how much pointing loss the sun sensor would lose. The safe mode profile make sure that the sun sensor is always pointed to the sun to continue providing power with a patch solar array while the main solar array being stowed.

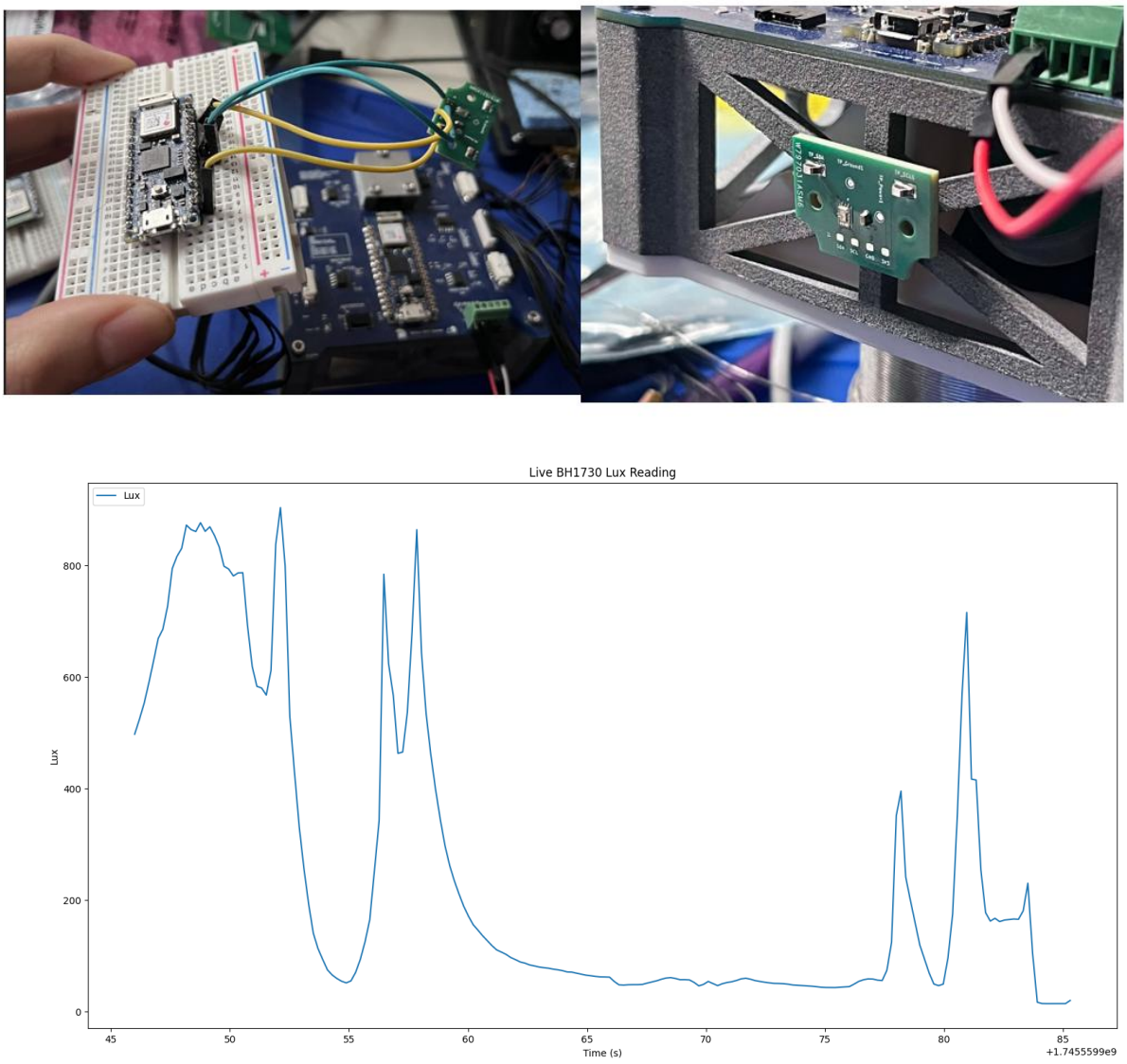


Figure 6.3 Unit test of digital sun sensor setup

Figure 6.4 provide the unit test setup of the RM3100 magnetometer sun sensor. The magnetometer will be separately mounted on the Cubesat but will still be routed to the ADCS Ventura board for feedback and control. The reason is that placing RM3100 near high-speed electronics causing signal integrity as well as magnetic flux to enter the sensor which introduces noises and more complex filtering techniques required.

The characterization of the sun sensor is collected at the systems level to observed the sensor fusion between the IMU and magnetometer in figure 6.7

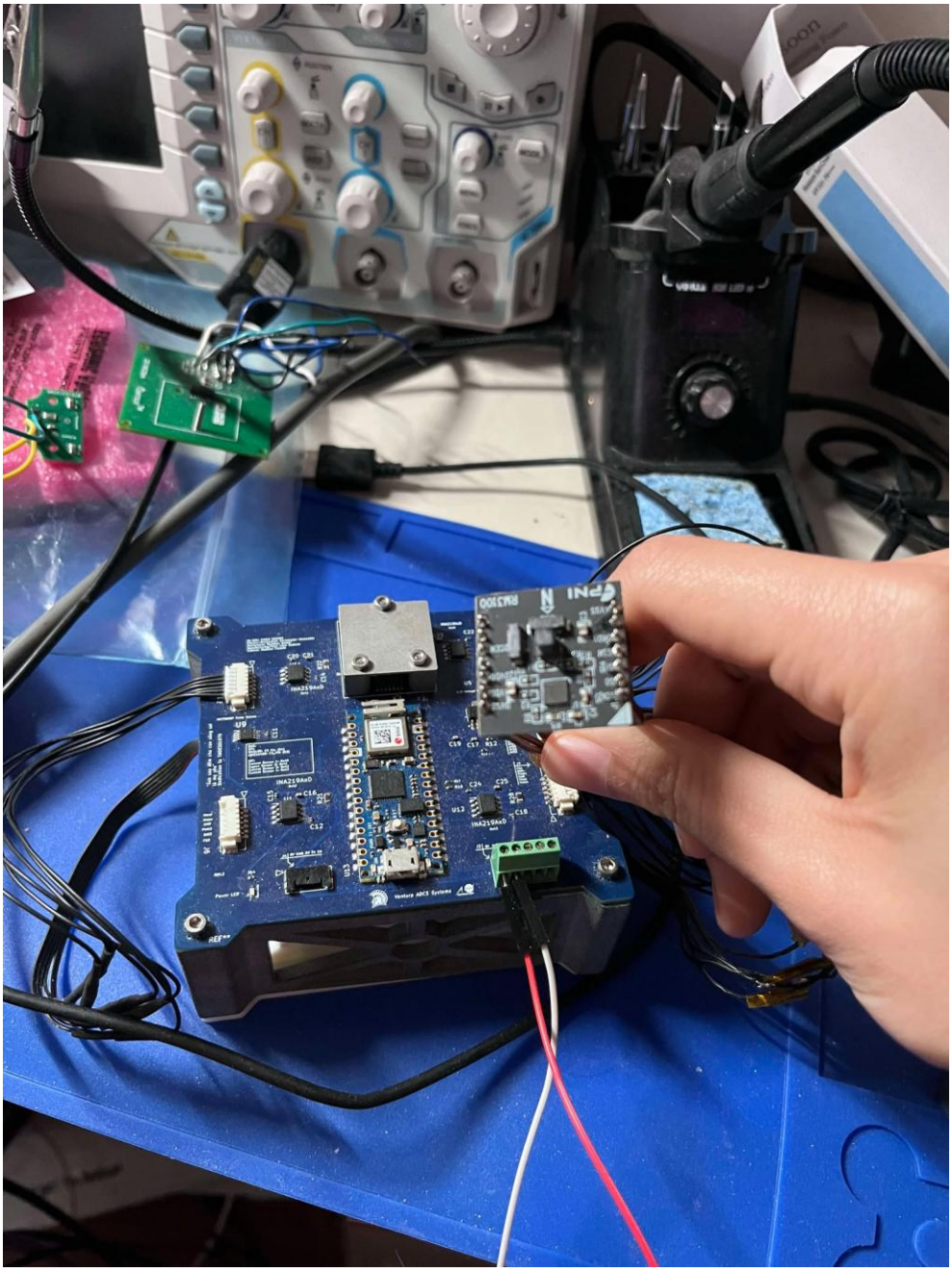


Figure 6.4 RM3100 magnetometer unit test

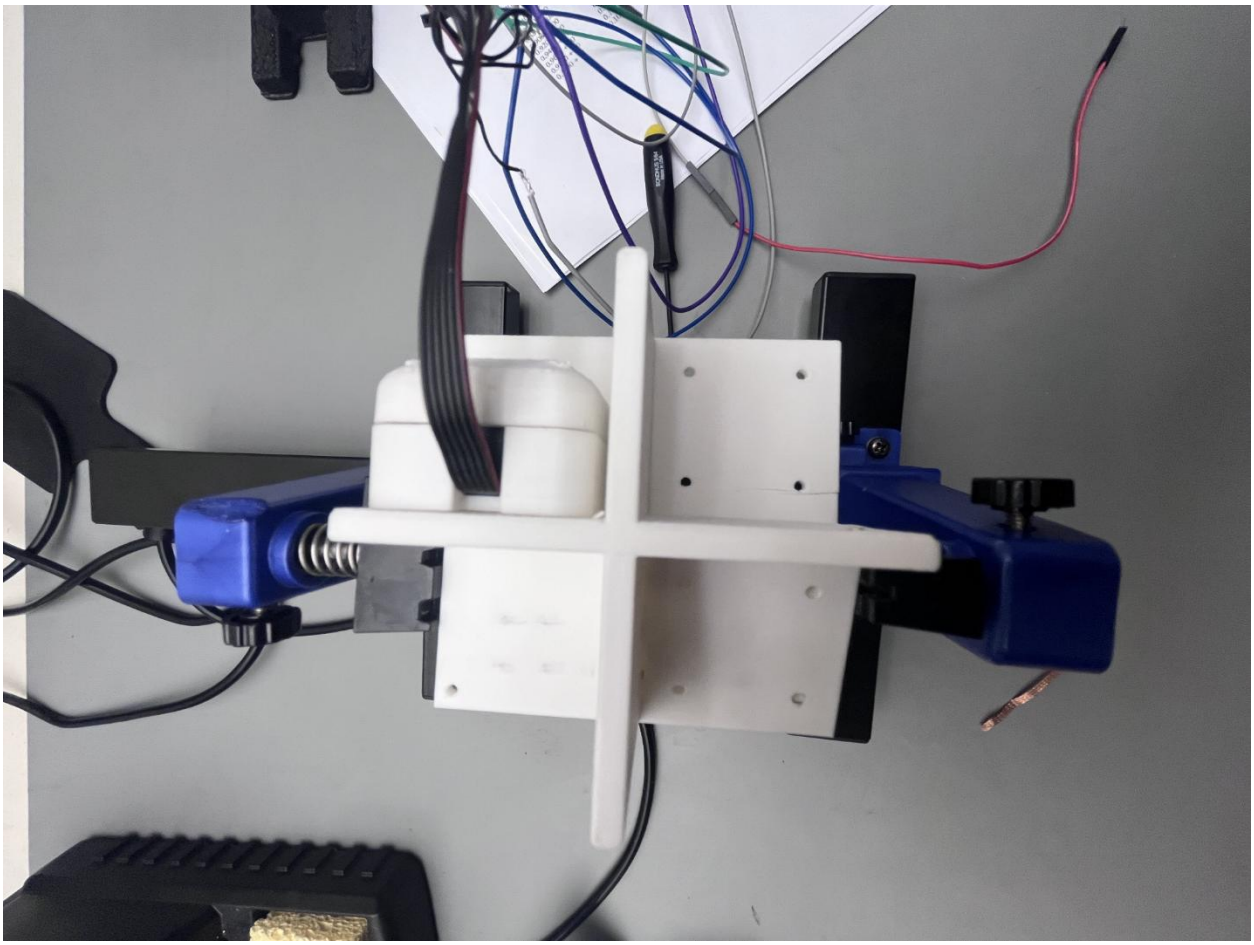


Figure 6.5. Testbed motion simulator 3 degree of freedoms

The unit under test would be the entire ADCS systems shown in Figure 6.5 such that the clamp on two sides are clamping on the two narrow features of the baseplate subsystem which hold the motors. The structure of the motion simulator is made up of rigid arms where ach rung allow independent rotation about one axis. There is a high precision rotary bearing with low friction which increases the smoothness of the operation. The stand is manually adjustable with kinematics screws adopted from optomechanical concepts/ The position feedback consists of a rotary encoder on each axis for a closed loop control. The range of motion would be 180 degrees for yaw pitch , 360 degrees for roll. For the control interface, the data will be through serial and Serial Peripheral Interface [SPI] interface with real time command input and feedback logging. The cable management would need to be wrapped to prevent tangling when there are mutli-turn rotation.

6.2 Controlling Loop

The control loop of the Hardware in the loop starts with reading the true spacecraft motion from the reaction wheels actuating the body. As for the gimbal integration it reads the real motion as the gimbal rotate in response to the ADCS commands. The script also use the date to update the orientation of the spacecraft's orientation in quaternion form. Then it integrates the gyro rate over time. Then it also normalize quaternions to prevent drifting effect.

For the Q-ref which is the desired orientation it is calculated based on the mission mode state machine such as a point to earth, point to Sun or pointing at a coordinate on earth. It also calculates the quaternion error by multiplying matrix of the current operations and the desired orientation to see the error as an output to feed into the PD controller. At this point the IMU also measures the reaction rate as the gimbal rotates the ADCS board.

The PD controller then generates torque commands of Tx Ty and Tz to individual wheel commands. It also uses pseudo inverse math to distribute the torque among multiple 3x4 reaction wheels. The scaling factor of the hardware must also be accounted for PWM operation. The motor output will be made up of speed and direction. Such speed and directions will be captured as feedback from the internal hall sensor. INA219 current sensor also collects the current drawn by each motor to convert into th actual torque's constant for feedback and monitoring. This essentially completes the closed for loop Hardware in the loop interface.

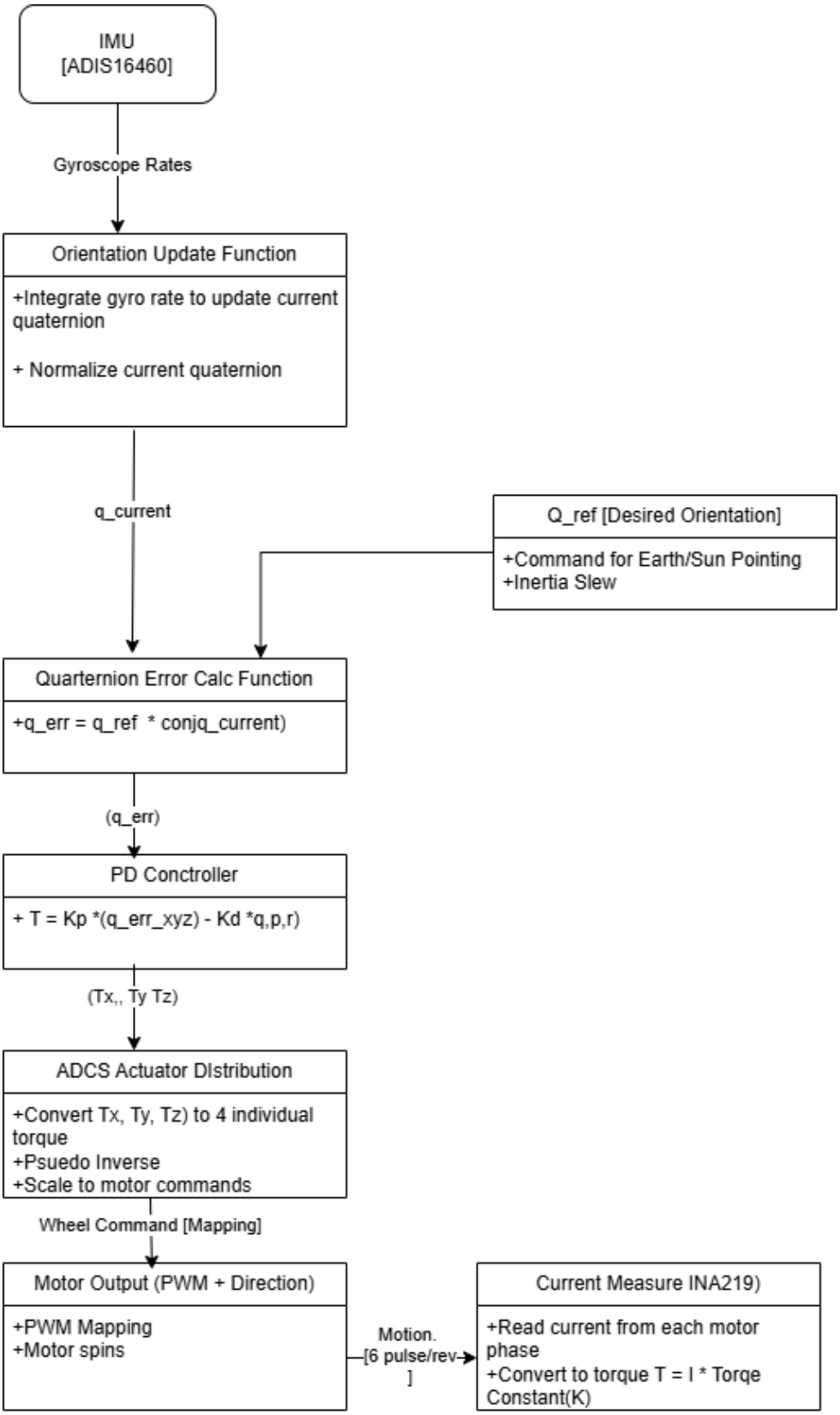


Figure 6.6 Control loop

6.3 Data Collection/Display

The data collection of the HWIL is streamed by telemetry viewer on a hosted local machine. Telemetry viewer is an app written in JavaScript to serve as a GUI interface. That GUI interface allows the user to set baud rate speed streamline the type of data type of the telemetry about to collect as well as the screen display for streaming telemetry.

Figure 6.7 contains 3x3 subplots of all sampling data vs their respective units. In position row one column number one, IMU ADIS16460's gyro data is plotted which can be observed to be nominal when gimbling the ADCS. The row one column 2 subplot is the verification of the redundant IMU on the Arduino. This IMU along with the RM3100 magnetometer and ADIIS16460 complete the sensor fusion. The IMU temperature are the average of both functions from the MCP9809 and the ADIS16460;s internal temperature sensor. Position row 2, column 1 displays the ADIS16460 IMU where it is nudged manually for acceleration. Z acceleration remains 1 m/s^2 due to the uncalibrated factor on testing day. Only 3 motors were being used that day/ This can be observed when torque B remains at 0 for all of the sampling. This was the test to run compensation from other reaction wheel.

This maneuver was performed when trying to point at the sun to simulate that reaction wheel B has voltage spike and forced it to shut down. So, torque demands from motor A was defaulted to be the one to compensate for torque B.

The bottom first two graphs are the output from the RM3100 magnetometer, The first one shows the individual axis output in micro Tesla. The second graph shows the magnitude combination of all 3 axis commands. As observed by the first 4100 samples, the RM3100 was being put closer to the digital Ventura board causing a spike in the magnetic detection field. To solve this problem, the decision was to isolate the sun sensor on the solar array where magnetic field from electronics would not interfere with the collection of data.

The last subplot is the total power consumed and drawn from the reaction wheel. This subplot also verify that motor B was off and not running to simulate anomaly. Motor A also consumed the most voltage as its duty cycle increased to compensate for motor B's failure.

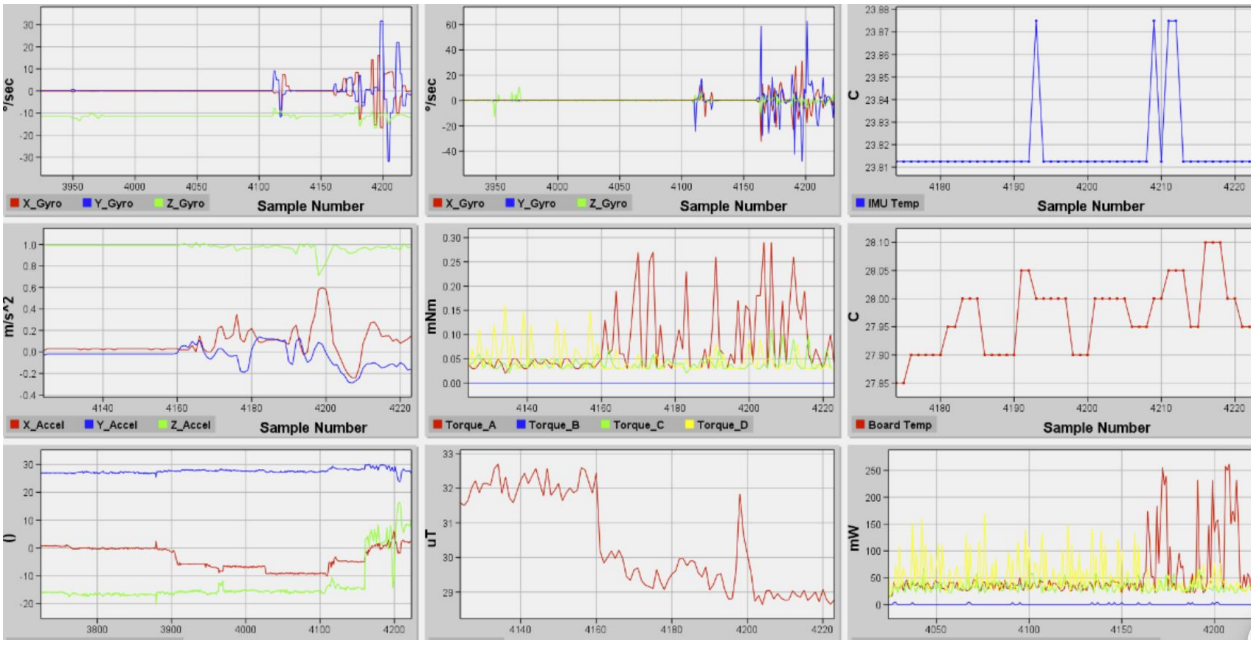


Figure 6.7. ADCS Command Dashboard via Telemetry Viewer

The IMU data was also collected and streamed online via an app called Grafana. The workflow of Grafana allows the user to collect and store database in InfluxDB, this can then be queried into displaying the data via Wi-Fi protocol. In the middle of the screen shows the "RSSI Plot" such that the higher the number the better the connections of the local Wi-Fi that is being communicated to the Arduino. The temperature collected from IMU also matches the local HWIL setup on the tester's machine host. The angular velocity around the principal axes were aggressively moved around to simulate the detumbling of the satellite in the worst-case scenario. The limits thresh hold, anything above 4000 mm/s linear velocity resulted in a declaration of losing the spacecraft.

Grafana

Figure 6.8. IMU Command Dashboard via Grafana

Overall, the hardware in the loop experiment ran smoothly many troubleshooting efforts. One effort to troubleshoot was the voltage drop of the cable when changing the setup. This causes the motor to not pass self-test which declare the anomaly of the ADCS wheel. This is a single point of failure that was fixed by doing configuration management and properly labeling the hardware and connectors before integration of every test case and test scenarios. There was also a problem collecting the sun sensor data in which it also self-test because it did not receive enough lux at 80 degrees off boresight. This was fixed by introducing bias and making an adaptive threshold and lowering the specifications for the self-test to passed.

References

[1] Fritz, M., Shoer, J., Singh, L., Henderson, T., McGee, J., and Rose, R., “Attitude Determination and Control System Design for the CYGNSS Microsatellite,” 2015.

[2] Gumiela, M., Sobiecki, M., Sażyński, K., Ambroszkiewicz, G., Kaczor, B., Polak, S., Gajoch, G., and Roszkowski, D., “PW-SAT2 Critical Design Review,” Warsaw, November 2016.

[3] “Spacecraft Sun Sensors, NASA Space Vehicle Design Criteria /Guidance and Control/,” June 1970.

[4] Yousefian, P., Durali, M., and Jalali, M. A., “Optimal Design and Simulation of Sensor Arrays for Solar Motion Estimation,” IEEE Sensors Journal, Vol. 17, No. 6, 2017, pp. 1673–1680.

[5] Nidey, R. A., “Aspect Sensing, Astrostat Design and Orientation Control in Space Research.” November 1959.

[6] Bollattino, S., and Stesina, F., “Fast Development and Validation of a Sensing Suite System for CubeSats,” Acta Astronautica, Vol. 223, 2024, pp. 342–354. https://doi.org/10.1016/j.actaastro.2024.07.005

[7] Zhu, J., Zhou, H., Wang, Z., and Yang, S., “Improved Multi-Sensor Fusion Positioning System Based on GNSS/LiDAR/Vision/IMU With Semi-Tight Coupling and Graph Optimization in GNSS Challenging Environments,” IEEE Access, Vol. 11, 2023, pp. 95711–95723. https://doi.org/10.1109/ACCESS.2023.3311359

[8] Shim, H., Kim, O. J., Park, M., Choi, M., and Kee, C., “Development of Hardware-in-the-Loop Simulation for CubeSat Platform: Focusing on Magnetometer and Magnetorquer,” IEEE Access, Vol. 11, 2023, pp. 73164–73179. https://doi.org/10.1109/ACCESS.2023.3294565

[9] Markley, F. L., and Crassidis, J. L., “Fundamentals of Spacecraft Attitude Determination and Control,” Springer, New York, 2014.

[10] NASA, & Starin, S. R. 19.1 Attitude Determination and Control Systems, pp. 14

[11] Ismail, Z., and Varatharajoo, R., “REACTION WHEEL CONFIGURATIONS FOR HIGH AND MIDDLE ORBITS,” ARPN Journal of Engineering and Applied Sciences, Vol. 10, No. 21, 2015. Retrieved 1 October 2024. https://www.arpnjournals.org/jeas/research_papers/rp_2015/jeas_1115_3023.pdf

[12] Wafi, I. El, Haloua, M., Guennoun, Z., and Moudden, Z., “Constrained Attitude Control of a Cubesat with Redundant Reaction Wheel Configurations.”

[13] Weston, S. V, Burkhard, C. D., Stupl, J. M., Ticknor, R. L., Yost, B. D., Austin, R. A., Galchenko, P., Newman, L. K., and Santos Soto, L., “State-of-the-Art Small Spacecraft Technology,” March 2025.

[14] Krishna, N. S., Gosavi, S., Singh, S., Saxena, N., Kailaje, A., and Datla, V., “Design and Implementation of a Reaction Wheel System for CubeSats,” 2018.

[15] Oberg, Jones, M., and Horton, Machinerys Handbook, New York: Industrial Press, Incorporated, 2004.

[16] Y. -S. Lee, N. Xuan-Mung, T. -D. Do, J. -Y. Park, C. -H. Lee and S. K. Hong, "Design of a Nanosatellite Attitude Control Hardware-in-the-loop Simulation System," 2023, pp. 609-613. 10.1109/ICSSE58758.2023.10227144

[17] Gaber, K., El_Mashade, M. B., and Aziz, G. A., “Hardware-in-the-loop real-time validation of micro-satellite Attitude Control,” Computers & Electrical Engineering, vol. 85, Jul. 2020, p. 106679.

[18] Cepeda, J. G., Jeremiah, R., “Thermal and structural electronic packaging analysis for space and Extreme Environments”, S.l.: CRC PRESS, 2024.

[19] California Polytechnic State University, “CubeSat Design Specification”, 2021, Retrieved 15 September 2024. https://static1.squarespace.com/static/5418c831e4b0fa4ecac1ba cd/t/56e9b62337013b6c063a655a/1458157095454/cds_rev13_final2.pdf.

[20] Y. Q. J. Zhang, "Project of 1 DoF Attitude Control System of 1U," Technical University of Catalonia — BarcelonaTech, Barcelona, 2021.

[21] Merz P., Reimer K., and Schwarzelbach O. "Combined MEMS inertial sensors for IMU applications" 24 January 2010. doi: 10.1109/MEMSYS.2010.5442458

[22] Grewal, M. S., “How good is your gyro [ask the experts],” vol. 30, Feb. 2010, pp. 12–86.

[23] Rohm, “BH1730FVC,” ROHM. Retrieved 15 September 2024. https://www.rohm.com/products/sensors-mems/ambient-light-sensor-ics/digital-16bit-serial-output/bh1730fvc-product.

[24] Faulhaber, "BLDC FB2610C". Retrieved 15 September 2024. https://www.faulhaber.com/en/products/series/2610b-sc/

[25] Rawasdeh S., " CubeSat Aerodynamic Stability at ISS Altitude and Inclination". 13 August 2012

[26] Wertz, J. R., and Larson, W. J., Space mission analysis and design, Dordrecht: Kluwer, 1991.

[27] Felix Hebeler (2024). Solar Radiation (https://www.mathworks.com/matlabcentral/fileexchange/19791-solar-radiation), MATLAB Central File Exchange. Retrieved 10 December 2024.

[28] Wong, E. C., and Breckenridge, W. G., “Inertial attitude determination for a dual-spin planetary spacecraft,” Journal of Guidance, Control, and Dynamics, vol. 6, Nov. 1983, pp. 491–498.

[29] Srivastava, R., Kulkarni, N. A., and Kumar, V. S., “Designing an effective attitude determination and control system (ADCS) for CubeSat and analysis of the effects of disturbance forces,” Jul 2024.

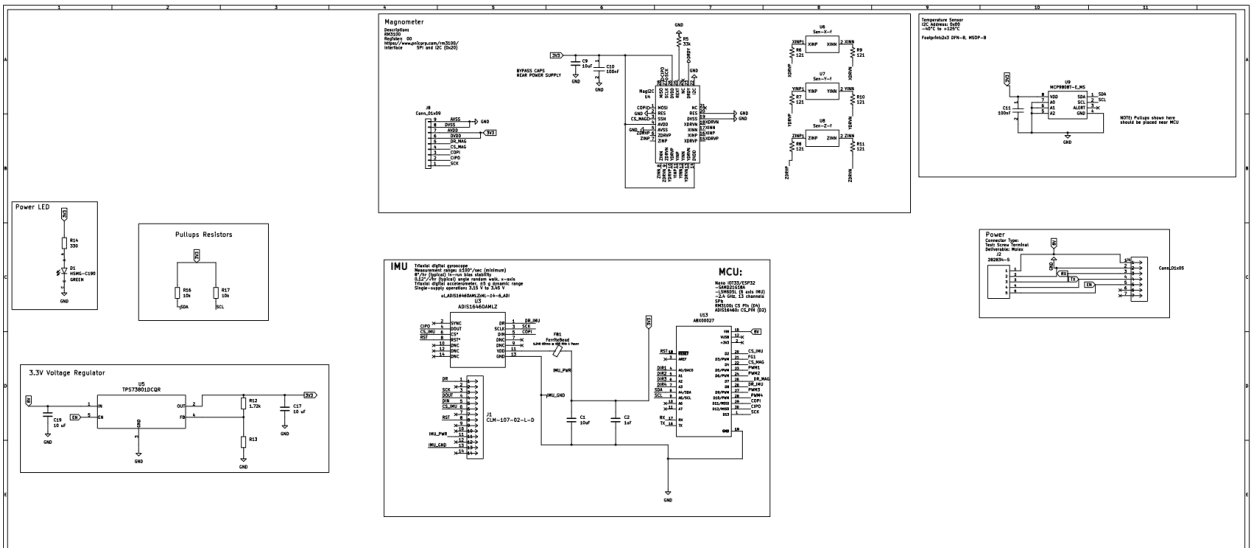
[30] Umich, “DC Motor Speed: Simulink modeling,” *Control Tutorials for MATLAB and Simulink - Motor Speed: Simulink Modeling*. Retrieved 20 January 2025.

https://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed&section=SimulinkModeling.

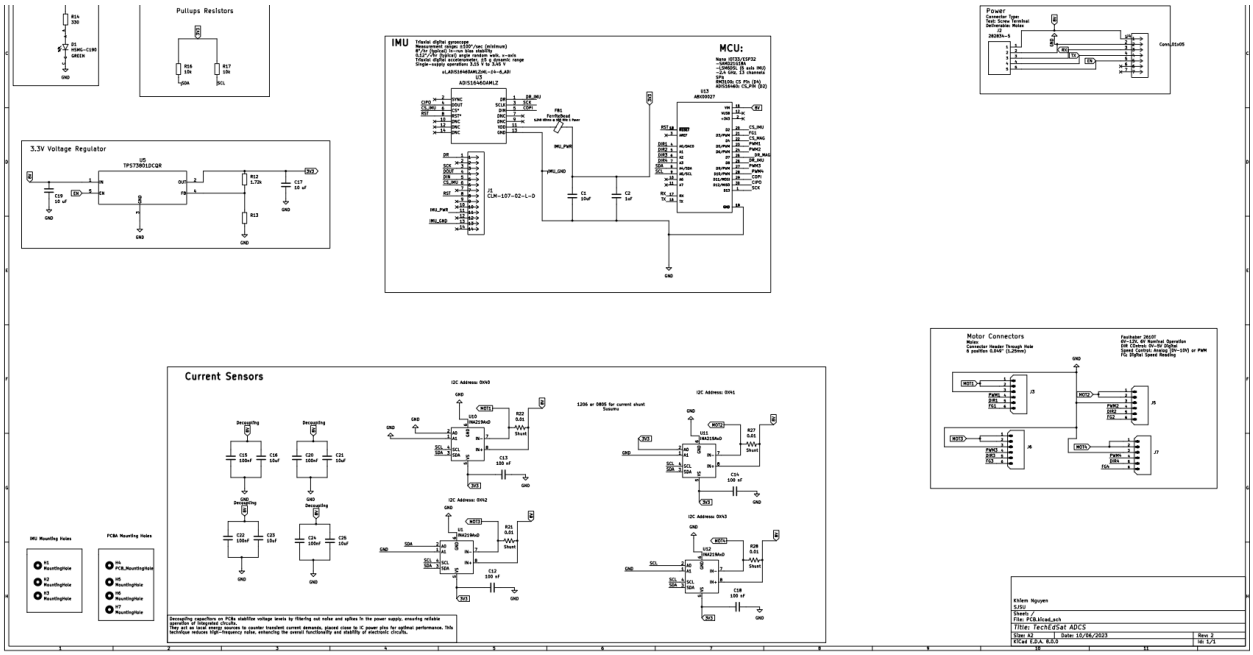
Appendices

Appendix A: ADCS Ventura Schematic/Layout

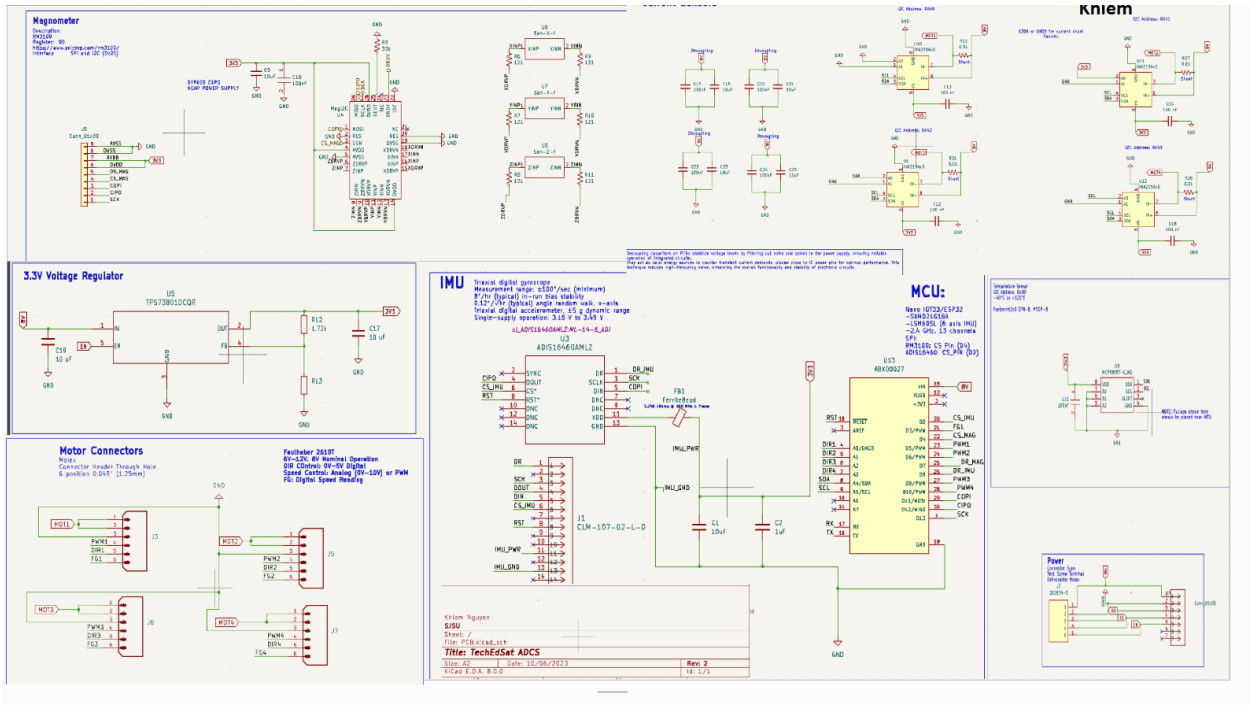
A.1 Top Half of Schematic Drawing



A.2 Bottom Half of Schematic Drawing



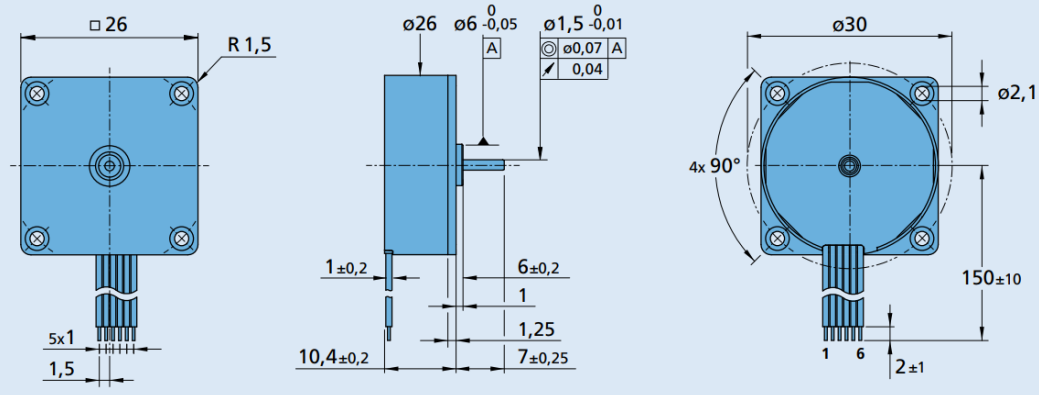
A.3 Schematic Overall



Appendix B: BLDC Motor

B.1 Mechanical Interface

Dimensional drawing



2610 T ... B SC

Option, cable and connection information

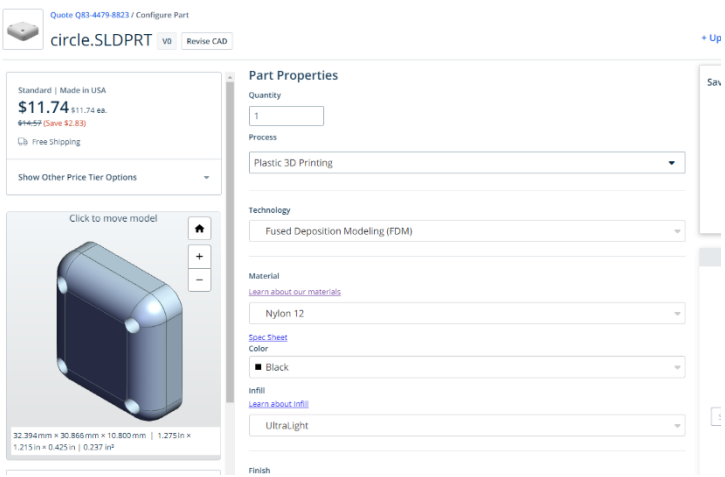
B.2 Specifications Parameters

2610 ... B SC

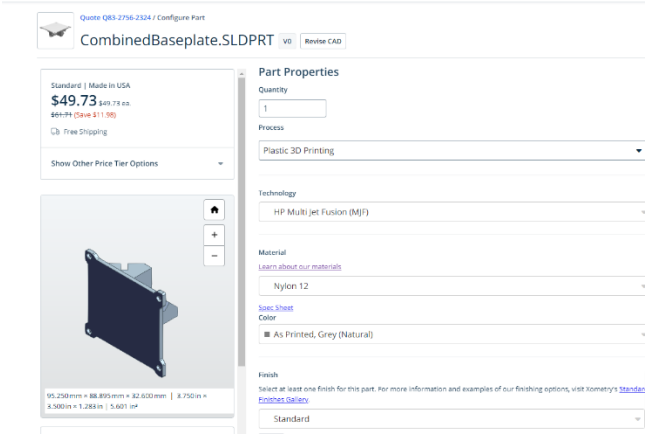
Values at 22°C and nominal voltage	2610 T		006 B SC	012 B SC	
Power supply electronic	U_P		4 ... 18	4 ... 18	V DC
Power supply motor	U_{mot}		1,7 ... 18	1,7 ... 18	V DC
Nominal voltage for motor	U_N		6	12	V
No-load speed (at U_N)	n_0		6 700	6 650	min^{-1}
Peak torque (S2 operation for max. 2s/1s)	$M_{max.}$		6	6	mNm
Torque constant	k_M		8,8	17,6	mNm/A
PWM switching frequency	f_{PWM}		96	96	kHz
Efficiency electronic	η		95	95	%
Standby current for electronic (at U_N)	I_{el}		0,02	0,02	A
Speed range (up to 12V / 18V)			400 ... 13 300	400 ... 10 000	min^{-1}
Shaft bearings		ball bearings, preloaded			
Shaft load max.:					
– with shaft diameter		1,5			mm
– radial at 3 000 min^{-1} (3 mm from mounting flange)		4			N
– axial at 3 000 min^{-1} (push only)		3,5			N
– axial at standstill (push only)		17,5			N
Shaft play:					
– radial		≤ 0,015			µm
– axial		= 0			µm
Operating temperature range		-25 ... +80			°C
Housing material		plastic			
Mass		20,1			g

Appendix C : Intergation and Manufactuuring

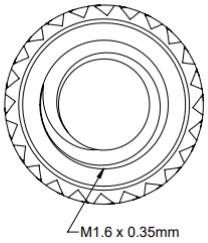
C.1 Nylon 12 Motor Cap

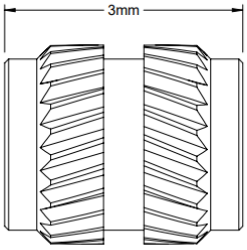


C.2 Baseplate



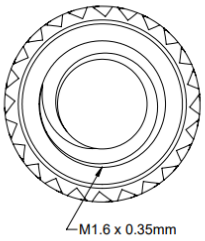
C.3 M1.6 Class 6H Brass Threadcerts

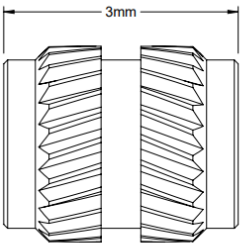




For Minimum Material Thickness: 3.77mm
Drill Bit Size: No. 44
For Maximum Hole Diameter: 2.2mm

McMASTER-CARR CAD PART NUMBER 92120A150
http://www.mcmaster.com
© 2023 McMaster-Carr Supply Company
Information in this drawing is provided for reference only.
Brass Heat-Set
Inserts for Plastic

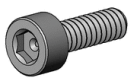


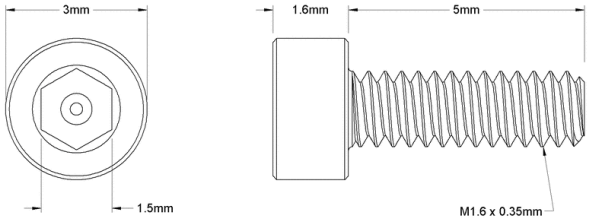


For Minimum Material Thickness: 3.77mm
Drill Bit Size: No. 44
For Maximum Hole Diameter: 2.2mm

McMASTER-CARR CAD PART NUMBER 92120A150
http://www.mcmaster.com
© 2023 McMaster-Carr Supply Company
Information in this drawing is provided for reference only.
Brass Heat-Set
Inserts for Plastic

C.4 Motor Integration Screw

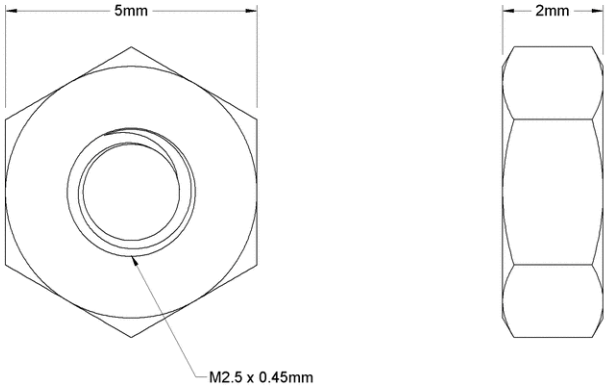


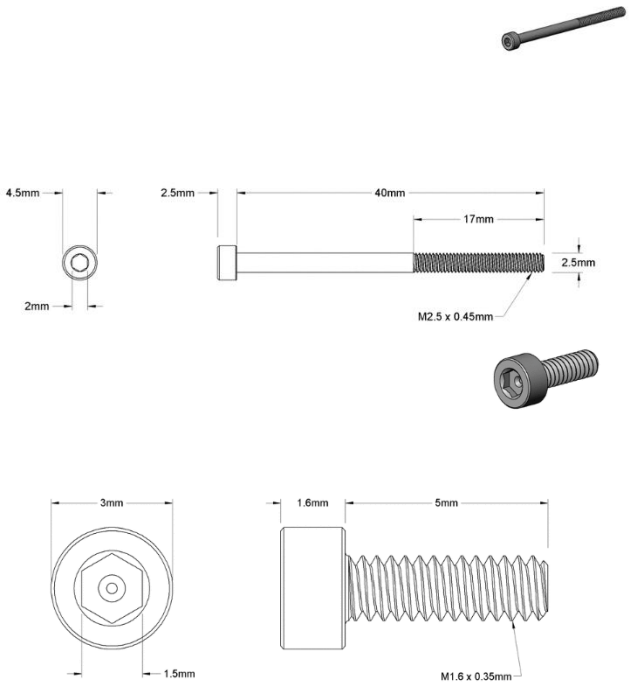


McMASTER-CARR® CAD
http://www.mcmaster.com
© 2023 McMaster-Carr Supply Company
Information in this drawing is provided for reference only.

PART NUMBER 91290A022

Alloy Steel Socket Head Screw





McMASTER-CARR CAD PART NUMBER 91290A022

http://www.mcmaster.com

Alloy Steel Socket Head Screw

Appendix D : Telemtry Code

D.1 Main File to Call other files

```
#include <Arduino.h>
#include "ADIS16460m.h"
#include "INA219m.h"
#include "LSMm.h"
#include "Motor_ALVm.h"
#include "RM3100_Officialm.h"
#include "MCP9808.h"

// put function declarations here:

void setup() {
  // put your setup code here, to run once:
  ADIS16460_setup();
  INA219_setup();
  LSM_setup();
  //Motor_ALV_setup();
  RM3100_setup();
  MCP9808_setup();
}

void loop() {
  // put your main code here, to run repeatedly:
  ADIS16460_loop();
  INA219_loop();
  LSM_loop();
  //Motor_ALV_loop();
  RM3100_loop();
  MCP9808_loop();
}
```

D.2 ADIS16460 Code

```
#include <SPI.h>
/*Connection

Arduino Nano ESP32      -->        ADIS16460
VCC 3.3V Out            -->        VDD (11)
GND                     -->        GND (9)
RST(RST)                 -->        RST (1)
SCK(D13)                -->        SCK  (2)
CS(D10)                 -->        CS(3)
CIPO/MISO (D12)         -->        DOUT(4)
COPI/MOSI (D11)         -->        DIN (6)
DR(D8)                  -->        DR(13)
*/

// ADIS16460 Register Definitions
#define FLASH_CNT   0x00  //Flash memory write count
#define DIAG_STAT   0x02  //Diagnostic and operational status
```

```
#define X_GYRO_LOW  0x04  //X-axis gyroscope output, lower word
#define X_GYRO_OUT  0x06  //X-axis gyroscope output, upper word
#define Y_GYRO_LOW  0x08  //Y-axis gyroscope output, lower word
#define Y_GYRO_OUT  0x0A  //Y-axis gyroscope output, upper word
#define Z_GYRO_LOW  0x0C  //Z-axis gyroscope output, lower word
#define Z_GYRO_OUT  0x0E  //Z-axis gyroscope output, upper word
#define X_ACCL_LOW  0x10  //X-axis accelerometer output, lower word
#define X_ACCL_OUT  0x12  //X-axis accelerometer output, upper word
#define Y_ACCL_LOW  0x14  //Y-axis accelerometer output, lower word
#define Y_ACCL_OUT  0x16  //Y-axis accelerometer output, upper word
#define Z_ACCL_LOW  0x18  //Z-axis accelerometer output, lower word
#define Z_ACCL_OUT  0x1A  //Z-axis accelerometer output, upper word
#define SMPL_CNTR   0x1C  //Sample Counter, MSC_CTRL[3:2]=11
#define TEMP_OUT    0x1E  //Temperature output (internal, not calibrated)
#define X_DELT_ANG  0x24  //X-axis delta angle output
#define Y_DELT_ANG  0x26  //Y-axis delta angle output
#define Z_DELT_ANG  0x28  //Z-axis delta angle output
#define X_DELT_VEL  0x2A  //X-axis delta velocity output
#define Y_DELT_VEL  0x2C  //Y-axis delta velocity output
#define Z_DELT_VEL  0x2E  //Z-axis delta velocity output
#define MSC_CTRL    0x32  //Miscellaneous control
#define SYNC_SCAL   0x34  //Sync input scale control
#define DEC_RATE    0x36  //Decimation rate control
#define FLTR_CTRL   0x38  //Filter control, auto-null record time
#define GLOB_CMD    0x3E  //Global commands
#define XGYRO_OFF   0x40  //X-axis gyroscope bias offset error
#define YGYRO_OFF   0x42  //Y-axis gyroscope bias offset error
#define ZGYRO_OFF   0x44  //Z-axis gyroscope bias offset factor
#define XACCL_OFF   0x46  //X-axis acceleration bias offset factor
#define YACCL_OFF   0x48  //Y-axis acceleration bias offset factor
#define ZACCL_OFF   0x4A  //Z-axis acceleration bias offset factor
#define LOT_ID1     0x52  //Lot identification number
#define LOT_ID2     0x54  //Lot identification number
#define PROD_ID     0x56  //Product identifier
#define SERIAL_NUM  0x58  //Lot-specific serial number
#define CAL_SGNTR   0x60  //Calibration memory signature value
#define CAL_CRC     0x62  //Calibration memory CRC values
#define CODE_SGNTR  0x64  //Code memory signature value
#define CODE_CRC    0x66  //Code memory CRC values
#define ADIS16460_CS   2  // Assuming Chip Select is connected to pin 10
#define ADIS16460_DR   8

const float GYRO_SCALE = 0.005;   // degrees per second per LSB
const float ACCEL_SCALE = 0.00025; // g's per LSB
const float deltaD = 0.005; // Multiply by delta angle scale (0.005 degrees/LSB)
const float deltaV = 2.5; // Multiply by velocity scale (2.5 mm/sec/LSB)

SPISettings adisSettings(1000000, MSBFIRST, SPI_MODE3);  // 1MHz, MSB first, mode
3

// Function declarations
int16_t readRegister(uint8_t regAddress);
```

```
int regWrite(uint8_t regAddr, int16_t regData);

void ADIS16460_setup() {
  Serial.begin(19200);
  while(!Serial);

  SPI.begin();
  delay(100);
  pinMode(ADIS16460_CS, OUTPUT);
  pinMode(ADIS16460_DR, INPUT);
  digitalWrite(ADIS16460_CS, HIGH);
  digitalWrite(ADIS16460_DR, HIGH);

  // Additional initialization if necessary...
  regWrite(MSC_CTRL, 0xC1);  // Enable Data Ready, set polarity
  delay(20);
  regWrite(FLTR_CTRL, 0x500); // Set digital filter
  delay(20);
  regWrite(DEC_RATE, 0), // Disable decimation
  delay(20);
}

void ADIS16460_loop() {
  int16_t x_gyro, y_gyro, z_gyro;
  int16_t x_accel, y_accel, z_accel;
  int16_t x_dangle, y_dangle, z_dangle;
  int16_t x_dvel, y_dvel, z_dvel;
  int16_t temp;

  x_gyro = readRegister(X_GYRO_OUT);
  y_gyro = readRegister(Y_GYRO_OUT);
  z_gyro = readRegister(Z_GYRO_OUT);

  x_accel = readRegister(X_ACCL_OUT);
  y_accel = readRegister(Y_ACCL_OUT);
  z_accel = readRegister(Z_ACCL_OUT);

  x_dangle = readRegister(X_DELT_ANG);
  y_dangle = readRegister(Y_DELT_ANG);
  z_dangle = readRegister(Z_DELT_ANG);

  x_dvel = readRegister(X_DELT_VEL);
  y_dvel = readRegister(Y_DELT_VEL);
  z_dvel = readRegister(Z_DELT_VEL);

  temp = readRegister(TEMP_OUT);

  Serial.print(x_gyro*GYRO_SCALE); Serial.print(",");
  Serial.print(y_gyro*GYRO_SCALE); Serial.print(",");
  Serial.print(z_gyro*GYRO_SCALE); Serial.print(",");
  Serial.print(x_accel*ACCEL_SCALE); Serial.print(",");
  Serial.print(y_accel*ACCEL_SCALE); Serial.print(",");
```

```
  Serial.print(z_accel*ACCEL_SCALE); Serial.print(",");
  Serial.print(x_dangle*deltaD); Serial.print(",");
  Serial.print(y_dangle*deltaD); Serial.print(",");
  Serial.print(z_dangle*deltaD); Serial.print(",");
  Serial.print(x_dvel*deltaV); Serial.print(",");
  Serial.print(y_dvel*deltaV); Serial.print(",");
  Serial.print(z_dvel*deltaV); Serial.print(",");

  //Serial.println((temp*0.05)+25);
  Serial.print((temp*0.05)+25); Serial.print(",");

  //delay(10);  // delay for 1 seconds
}

int16_t readRegister(uint8_t regAddress) {

  SPI.beginTransaction(adisSettings);

  digitalWrite(ADIS16460_CS, LOW);
  SPI.transfer(regAddress);  // Shift left to accommodate the R/W bit
  SPI.transfer(0x00);
  digitalWrite(ADIS16460_CS, HIGH);

  delayMicroseconds(40);

  digitalWrite(ADIS16460_CS, LOW);



  uint8_t msbData = SPI.transfer(0x00); // Send (0x00) and place upper byte into
variable
  uint8_t lsbData = SPI.transfer(0x00); // Send (0x00) and place lower byte into
variable
  digitalWrite(ADIS16460_CS, HIGH);

  delayMicroseconds(40); // Delay to not violate read rate (16 us)

  SPI.endTransaction();

  int16_t dataOut = (msbData << 8) | (lsbData & 0xFF); // Concatenate upper and
lower bytes

  return (dataOut);
}

int regWrite(uint8_t regAddr, int16_t regData) {

  // Write register address and data
  uint16_t addr = (((regAddr & 0x7F) | 0x80) << 8); // Toggle sign bit, and check
that the address is 8 bits
  uint16_t lowWord = (addr | (regData & 0xFF)); // OR Register address (A) with
data(D) (AADD)
```

```
  uint16_t highWord = ((addr | 0x100) | ((regData >> 8) & 0xFF)); // OR Register
address with data and increment address

  // Split words into chars
  uint8_t highBytehighWord = (highWord >> 8);
  uint8_t lowBytehighWord = (highWord & 0xFF);
  uint8_t highBytelowWord = (lowWord >> 8);
  uint8_t lowBytelowWord = (lowWord & 0xFF);

  // Write highWord to SPI bus
  digitalWrite(ADIS16460_CS, LOW); // Set CS low to enable device
  SPI.transfer(highBytelowWord); // Write high byte from low word to SPI bus
  SPI.transfer(lowBytelowWord); // Write low byte from low word to SPI bus
  digitalWrite(ADIS16460_CS, HIGH); // Set CS high to disable device

  delayMicroseconds(40); // Delay to not violate read rate (16 us)

  // Write lowWord to SPI bus
  digitalWrite(ADIS16460_CS, LOW); // Set CS low to enable device
  SPI.transfer(highBytehighWord); // Write high byte from high word to SPI bus
  SPI.transfer(lowBytehighWord); // Write low byte from high word to SPI bus
  digitalWrite(ADIS16460_CS, HIGH); // Set CS high to disable device

  return(1);
}
```

D.3 INA219 Code

```
#include <Wire.h>
#include <Adafruit_INA219.h>

//Motor Specifications
#define TORQUE_CONSTANT 8.8 // Torque constant in mNm/A
#define RATED_VOLTAGE 6     // Rated voltage in volts
#define SUPPLY_VOLTAGE 8    // Supply voltage in volts


#define PWM1 5
#define PWM2 6
#define PWM3 9
#define PWM4 10

#define DIR1 14
#define DIR2 15
#define DIR3 16
#define DIR4 17

Adafruit_INA219 ina219_A(0x40);
Adafruit_INA219 ina219_B(0x41);
Adafruit_INA219 ina219_C(0x42);
Adafruit_INA219 ina219_D(0x43);
```

```
void displaySensorData(Adafruit_INA219& sensor, const String& label);
void setMotorDirection(bool direction);
void stopAllMotors();
void runAllMotors(int speed);


void INA219_setup() {
  //Serial.begin(9600);
  while (!Serial) {
    delay(1); // Wait for serial port to connect
  }

  //Serial.println("Hello!");

  // Initialize the INA219 sensors
  if (!ina219_A.begin()) {
    Serial.println("Failed to find INA219 chip A");
    while (1) { delay(10); }
  }
  if (!ina219_B.begin()) {
    Serial.println("Failed to find INA219 chip B");
    while (1) { delay(10); }
  }
  if (!ina219_C.begin()) {
    Serial.println("Failed to find INA219 chip C");
    while (1) { delay(10); }
  }
  if (!ina219_D.begin()) {
    Serial.println("Failed to find INA219 chip D");
    while (1) { delay(10); }
  }

  //Serial.println("Measuring voltage and current with INA219 sensors ...");

  // Initialize motor control pins
  pinMode(PWM1, OUTPUT);
  pinMode(DIR1, OUTPUT);
  pinMode(PWM2, OUTPUT);
  pinMode(DIR2, OUTPUT);
  pinMode(PWM3, OUTPUT);
  pinMode(DIR3, OUTPUT);
  pinMode(PWM4, OUTPUT);
  pinMode(DIR4, OUTPUT);
}

void INA219_loop() {
  // Display sensor data for all INA219 sensors
  displaySensorData(ina219_A, "Sensor A");
  displaySensorData(ina219_B, "Sensor B");
  displaySensorData(ina219_C, "Sensor C");
  displaySensorData(ina219_D, "Sensor D");
```

```
  // Spin motors clockwise for 5 seconds
  setMotorDirection(HIGH);
  runAllMotors(250);
  //delay(5000);

  // Spin motors counterclockwise for 2 seconds
  //setMotorDirection(LOW);
  //runAllMotors(100);
  //delay(2000);

  // Stop all motors
  //stopAllMotors();
  //delay(2000);
}

void displaySensorData(Adafruit_INA219& sensor, const String& label) {
  float shuntvoltage = sensor.getShuntVoltage_mV();
  float busvoltage = sensor.getBusVoltage_V();
  float current_mA = sensor.getCurrent_mA();
  float power_mW = sensor.getPower_mW();
  float loadvoltage = busvoltage + (shuntvoltage / 1000);

    // Calculating torque in mNm
  float torque = TORQUE_CONSTANT * current_mA / 1000; // Convert current from mA
to A

  // Calculating true RPM
  float trueRPM = (busvoltage / RATED_VOLTAGE) *6700; // n0 is the no-load speed
at rated voltage

  Serial.print(torque); Serial.print(",");
  Serial.print(trueRPM); Serial.print(",");


  //Serial.print(label); Serial.print(" Bus Voltage: ");
  Serial.print(busvoltage); Serial.print(",");
  //Serial.println(" V");
  //Serial.print(label); Serial.print(" Shunt Voltage: ");
  Serial.print(shuntvoltage); Serial.print(",");
  //Serial.println(" mV");
  //Serial.print(label); Serial.print(" Load Voltage: ");
  Serial.print(loadvoltage); Serial.print(",");
  //Serial.println(" V");
  //Serial.print(label); Serial.print(" Current: ");
  Serial.print(current_mA); Serial.print(",");
  //Serial.println(" mA");
  //Serial.print(label); Serial.print(" Power: ");
  Serial.print(power_mW); Serial.print(",");
  //Serial.println(" mW");
  //Serial.println("");
}
```

```
void setMotorDirection(bool direction) {
  digitalWrite(DIR1, direction);
  digitalWrite(DIR2, direction);
  digitalWrite(DIR3, direction);
  digitalWrite(DIR4, direction);
}

void runAllMotors(int speed) {
  analogWrite(PWM1, speed);
  analogWrite(PWM2, speed);
  analogWrite(PWM3, speed);
  analogWrite(PWM4, speed);
}

void stopAllMotors() {
  analogWrite(PWM1, 0);
  analogWrite(PWM2, 0);
  analogWrite(PWM3, 0);
  analogWrite(PWM4, 0);
}
```

D.4 MCP9808 Code

```
/*************************************************************************/
/*!
This is a demo for the Adafruit MCP9808 breakout
----> http://www.adafruit.com/products/1782
Adafruit invests time and resources providing this open source code,
please support Adafruit and open-source hardware by purchasing
products from Adafruit!
*/
/*************************************************************************/

#include <Wire.h>
#include "Adafruit_MCP9808.h"

// Create the MCP9808 temperature sensor object
Adafruit_MCP9808 tempsensor = Adafruit_MCP9808();

void MCP9808_setup() {
  Serial.begin(9600);
  while (!Serial); //waits for serial terminal to be open, necessary in newer
arduino boards.
  //Serial.println("MCP9808 demo");

  // Make sure the sensor is found, you can also pass in a different i2c
  // address with tempsensor.begin(0x19) for example, also can be left in blank
for default address use
  // Also there is a table with all addres possible for this sensor, you can
connect multiple sensors
```

```
  // to the same i2c bus, just configure each sensor with a different address and
define multiple objects for that
  //  A2 A1 A0 address
  //  0  0  0   0x18  this is the default address
  //  0  0  1   0x19
  //  0  1  0   0x1A
  //  0  1  1   0x1B
  //  1  0  0   0x1C
  //  1  0  1   0x1D
  //  1  1  0   0x1E
  //  1  1  1   0x1F
  if (!tempsensor.begin(0x18)) {
    Serial.println("Couldn't find MCP9808! Check your connections and verify the
address is correct.");
    while (1);
  }

   //Serial.println("Found MCP9808!");

  tempsensor.setResolution(3); // sets the resolution mode of reading, the modes
are defined in the table bellow:
  // Mode Resolution SampleTime
  //  0    0.5°C       30 ms
  //  1    0.25°C      65 ms
  //  2    0.125°C     130 ms
  //  3    0.0625°C    250 ms
}

void MCP9808_loop() {
  //Serial.println("wake up MCP9808.... "); // wake up MCP9808 - power
consumption ~200 mikro Ampere
  tempsensor.wake();   // wake up, ready to read!

  // Read and print out the temperature, also shows the resolution mode used for
reading.
  //Serial.print("Resolution in mode: ");
  Serial.print(tempsensor.getResolution()); Serial.print(",");
  float c = tempsensor.readTempC();
  float f = tempsensor.readTempF();
  //Serial.print("Temp: ");
  Serial.print(c, 4); Serial.print(",");
  //Serial.print("*C\t and ");
  Serial.println(f, 4);
  //Serial.println("*F.");

  //delay(2000);
  //Serial.println("Shutdown MCP9808.... ");
  tempsensor.shutdown_wake(1); // shutdown MSP9808 - power consumption ~0.1 mikro
Ampere, stops temperature sampling
  //Serial.println("");
  //delay(200);
}
```

D.5 Motor BLDC2610 Command Code

```
#include <Wire.h>
#include <Adafruit_INA219.h>

Adafruit_INA219 ina219_A_ALV(0x40);
Adafruit_INA219 ina219_B_ALV(0x41);
Adafruit_INA219 ina219_C_ALV(0x48);
Adafruit_INA219 ina219_D_ALV(0x4c);

const int motorSpeedPin = 6; // The PWM pin for motor speed control
const int motorDirectionPin = 14; // The digital pin for motor direction control

void Motor_ALV_setup() {
  //Serial.begin(9600);
  while (!Serial) {
    delay(1); // Wait for serial port to connect
  }

  //Serial.println("Hello!");

  // Initialize the INA219 sensors
  if (!ina219_A_ALV.begin()) {
    Serial.println("Failed to find INA219 chip A");
    while (1) { delay(10); }
  }
  if (!ina219_B_ALV.begin()) {
    Serial.println("Failed to find INA219 chip B");
    while (1) { delay(10); }
  }

  //Serial.println("Measuring voltage and current with INA219 sensors ...");

  // Initialize motor control pins
  pinMode(motorSpeedPin, OUTPUT);
  pinMode(motorDirectionPin, OUTPUT);
}

void displaySensorData_ALV(Adafruit_INA219& sensor, const String& label) {
  float shuntvoltage = sensor.getShuntVoltage_mV();
  float busvoltage = sensor.getBusVoltage_V();
  float current_mA = sensor.getCurrent_mA();
  float power_mW = sensor.getPower_mW();
  float loadvoltage = busvoltage + (shuntvoltage / 1000);

  //Serial.print(label); Serial.print(" Bus Voltage: ");
  Serial.print(busvoltage); Serial.print(",");
  //Serial.println(" V");
```

```
  //Serial.print(label); Serial.print(" Shunt Voltage: ");
  Serial.print(shuntvoltage); Serial.print(",");
  //Serial.println(" mV");
  //Serial.print(label); Serial.print(" Load Voltage: ");
  Serial.print(loadvoltage); Serial.print(",");
  //Serial.println(" V");
  //Serial.print(label); Serial.print(" Current: ");
  Serial.print(current_mA); Serial.print(",");
  //Serial.println(" mA");
  //Serial.print(label); Serial.print(" Power: ");
  Serial.print(power_mW); Serial.print(",");
  //Serial.println(" mW");
  //Serial.println("");
}

void Motor_ALV_loop() {
  // Display sensor data
  displaySensorData_ALV(ina219_A_ALV, "Sensor A");
  displaySensorData_ALV(ina219_B_ALV, "Sensor B");

  // Motor control logic
  digitalWrite(motorDirectionPin, HIGH);
  analogWrite(motorSpeedPin, 50); // Set motor speed to maximum
  delay(5000);

  analogWrite(motorSpeedPin, 0); // Stop the motor
  delay(5000);

  digitalWrite(motorDirectionPin, LOW);
  analogWrite(motorSpeedPin, 50); // Set motor speed to maximum again
  delay(5000);

  analogWrite(motorSpeedPin, 0); // Stop the motor
  delay(5000);
}
```

D.6 RM3100 Command Code

```
#include <Arduino.h>
#include <SPI.h>

//pin definitions
#define PIN_DRDY 8 //Set pin D9 to be the Data Ready Pin
#define PIN_CS 4 //Chip Select (SS) is set to Pin 10

//internal register values without the R/W bit
#define RM3100_REVID_REG 0x36 // Hexadecimal address for the Revid internal
register
```

```
#define RM3100_POLL_REG 0x00 // Hexadecimal address for the Poll internal
register
#define RM3100_CMM_REG 0x01 // Hexadecimal address for the Continuous Measurement
Mode internal register
#define RM3100_STATUS_REG 0x34 // Hexadecimal address for the Status internal
register
#define RM3100_CCX1_REG 0x04 // Hexadecimal address for the Cycle Count X1
internal register
#define RM3100_CCX0_REG 0x05 // Hexadecimal address for the Cycle Count X0
internal register

//options
#define initialCC 200 // Set the cycle count
#define singleMode 0 //0 = use continuous measurement mode; 1 = use single
measurement mode
#define useDRDYPin 1 //0 = not using DRDYPin ; 1 = using DRDYPin to wait for data

/*********************************************
Arduino Nano ESP32/IOT33     -->        ADIS16460

SCK (D13)                -->        SCK (1)
MISO/COPI (D12)                  -->        MISO/CIPO (2)
MOSI/COPI (D11)                -->       MOSI/COPI (3)
CS (D10)              -->        SSN  (4)
DRDY (D)               -->        DRDY(5)

Power supply           -->  DVDD(12), AVDD(13)
Power Supply GND       -->  DVSS(14), AVSS(7)
*********************************************/

// Function definitions
uint8_t readReg(uint8_t addr);
void writeReg(uint8_t addr, uint8_t data);
void changeCycleCount(uint16_t newCC);

uint8_t revid;
uint16_t cycleCount;
float gain;

void RM3100_setup() {
 // pinMode(PIN_DRDY, INPUT);
  pinMode(PIN_CS, OUTPUT);
  digitalWrite(PIN_CS, HIGH);
  SPI.begin(); // Initiate the SPI library
  SPI.beginTransaction(SPISettings(1000000, MSBFIRST, SPI_MODE0));
  //Serial.begin(9600); //set baud rate to 9600
  delay(100);

  revid = readReg(RM3100_REVID_REG);

  //Serial.print("REVID ID = 0x"); //REVID ID should be 0x22
  //Serial.println(revid, HEX);
```

```
  changeCycleCount(initialCC); //change the cycle count; default = 200 (lower
cycle count = higher data rates but lower resolution)

  cycleCount = readReg(RM3100_CCX1_REG);
  cycleCount = (cycleCount << 8) | readReg(RM3100_CCX0_REG);

  //Serial.print("Cycle Counts = "); //display cycle count
  //Serial.println(cycleCount);

  gain = (0.3671 * (float)cycleCount) + 1.5; //linear equation to calculate the
gain from cycle count

  //Serial.print("Gain = "); //display gain; default gain should be around 75 for
the default cycle count of 200
  //Serial.println(gain);

  if (singleMode){
    //set up single measurement mode
    writeReg(RM3100_CMM_REG, 0);
    writeReg(RM3100_POLL_REG, 0x70);
  }
  else{
    // Enable transmission to take continuous measurement with Alarm functions
off
    writeReg(RM3100_CMM_REG, 0x79);
  }
}

void RM3100_loop() {
  long x = 0;
  long y = 0;
  long z = 0;
  uint8_t x2,x1,x0,y2,y1,y0,z2,z1,z0;

  //wait until data is ready using 1 of two methods (chosen in options at top of
code)
  if(useDRDYPin){
    while(digitalRead(PIN_DRDY) == LOW); //check RDRY pin
  }
  else{
    while((readReg(RM3100_STATUS_REG) & 0x80) != 0x80); //read internal status
register
  }

  //read measurements
  digitalWrite(PIN_CS, LOW);
  delay(100);
  SPI.transfer(0xA4);
  x2 = SPI.transfer(0xA5);
  x1 = SPI.transfer(0xA6);
  x0 = SPI.transfer(0xA7);
```

```
  y2 = SPI.transfer(0xA8);
  y1 = SPI.transfer(0xA9);
  y0 = SPI.transfer(0xAA);

  z2 = SPI.transfer(0xAB);
  z1 = SPI.transfer(0xAC);
  z0 = SPI.transfer(0);

  digitalWrite(PIN_CS, HIGH);

  //special bit manipulation since there is not a 24 bit signed int data type
  if (x2 & 0x80){
      x = 0xFF;
  }
  if (y2 & 0x80){
      y = 0xFF;
  }
  if (z2 & 0x80){
      z = 0xFF;
  }

  //format results into single 32 bit signed value
  x = (x * 256 * 256 * 256) | (int32_t)(x2) * 256 * 256 | (uint16_t)(x1) * 256 |
x0;
  y = (y * 256 * 256 * 256) | (int32_t)(y2) * 256 * 256 | (uint16_t)(y1) * 256 |
y0;
  z = (z * 256 * 256 * 256) | (int32_t)(z2) * 256 * 256 | (uint16_t)(z1) * 256 |
z0;

  //calculate magnitude of results
  double uT = sqrt(pow(((float)(x)/gain),2) + pow(((float)(y)/gain),2)+
pow(((float)(z)/gain),2));

  //display results
  //Serial.print("Data in counts:");
  //Serial.print("   X:");
  Serial.print(x); Serial.print(",");
  // Serial.print("   Y:");
  Serial.print(y); Serial.print(",");
  //Serial.print("   Z:");
  Serial.print(z); Serial.print(",");

  //Serial.print("Data in microTesla(uT):");
  //Serial.print("   X:");
  Serial.print((float)(x)/gain); Serial.print(",");
  //Serial.print("   Y:");
  Serial.print((float)(y)/gain); Serial.print(",");
  //Serial.print("   Z:");
  Serial.print((float)(z)/gain); Serial.print(",");

  //Magnitude should be around 45 uT (+/- 15 uT)
```

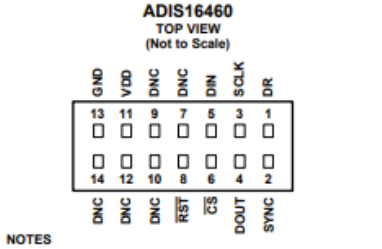
```
  //Serial.print("Magnitude(uT):");
  //Serial.println(uT);
  Serial.print(uT); Serial.print(",");
  //Serial.println();
}

//addr is the 7 bit value of the register's address (without the R/W bit)
uint8_t readReg(uint8_t addr){
  uint8_t data = 0;
  digitalWrite(PIN_CS, LOW);
  delay(100);
  SPI.transfer(addr | 0x80); //OR with 0x80 to make first bit(read/write bit)
high for read
  data = SPI.transfer(0);
  digitalWrite(PIN_CS, HIGH);
  return data;
}

//addr is the 7 bit (No r/w bit) value of the internal register's address, data
is 8 bit data being written
void writeReg(uint8_t addr, uint8_t data){
  digitalWrite(PIN_CS, LOW);
  delay(100);
  SPI.transfer(addr & 0x7F); //AND with 0x7F to make first bit(read/write bit)
low for write
  SPI.transfer(data);
  digitalWrite(PIN_CS, HIGH);
}

//newCC is the new cycle count value (16 bits) to change the data acquisition
void changeCycleCount(uint16_t newCC){
  uint8_t CCMSB = (newCC & 0xFF00) >> 8; //get the most significant byte
  uint8_t CCLSB = newCC & 0xFF; //get the least significant byte

  digitalWrite(PIN_CS, LOW);
  delay(100);
  SPI.transfer(RM3100_CCX1_REG & 0x7F); //AND with 0x7F to make first
bit(read/write bit) low for write
  SPI.transfer(CCMSB);  //write new cycle count to ccx1
  SPI.transfer(CCLSB);  //write new cycle count to ccx0
  SPI.transfer(CCMSB);  //write new cycle count to ccy1
  SPI.transfer(CCLSB);  //write new cycle count to ccy0
  SPI.transfer(CCMSB);  //write new cycle count to ccz1
  SPI.transfer(CCLSB);  //write new cycle count to ccz0
  digitalWrite(PIN_CS, HIGH);
}
```

Appendix E : Electronics Parts

E.1. IMU

E.1.1 IMU Pin Configuration

PIN CONFIGURATION AND FUNCTION DESCRIPTIONS



NOTES
1. THIS REPRESENTS THE PIN ASSIGNMENTS WHEN LOOKING DOWN AT THE CONNECTOR. SEE FIGURE 6.
2. MATING CONNECTOR: SAMTEC CLM-107-02 SERIES OR EQUIVALENT.
3. DNC = DO NOT CONNECT.

Figure 5. Pin Configuration

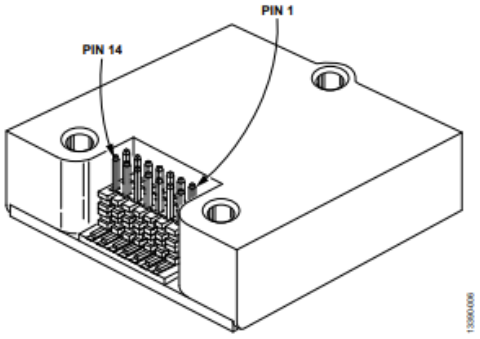


Figure 6. Pin Locations

Table 5. Pin Function Descriptions

Pin No.	Mnemonic	Type	Description
1	DR	Output	Data Ready Indicator.
2	SYNC	Input/Output	External Sync Input/Output, per MSC_CTRL. See Table 50.
3	SCLK	Input	SPI Serial Clock.
4	DOUT	Output	SPI Data Output. This pin clocks the output on the SCLK falling edge.
5	DIN	Input	SPI Data Input. This pin clocks the input on the SCLK rising edge.
6	$\overline{CS}$	Input	SPI Chip Select.
7	DNC	Not applicable	Do Not Connect. Do not connect to this pin.
8	$\overline{RST}$	Input	Reset.
9	DNC	Not applicable	Do Not Connect. Do not connect to this pin.
10	DNC	Not applicable	Do Not Connect. Do not connect to this pin.
11	VDD	Supply	Power Supply.
12	DNC	Not applicable	Do Not Connect. Do not connect to this pin.
13	GND	Supply	Power Ground.
14	DNC	Not applicable	Do Not Connect. Do not connect to this pin.

E1.1.2 IMU Application

APPLICATIONS INFORMATION

MOUNTING TIPS

The ADIS16460 package supports installation onto a printed circuit board (PCB) or rigid enclosure, using three M2 or 2-56 machine screws, using a torque that is between 20 inch ounces and 40 inch ounces. When designing a mechanical interface for the ADIS16460, avoid placing unnecessary translational stress on the electrical connector because it can influence the bias repeatability behaviors of the inertial sensors. When the same PCB also has the mating connector, the use of passthrough holes for the mounting screws may be required. Figure 33 shows a detailed view of the PCB pad design when using one of the connector variants in the CLM-107-02 family.

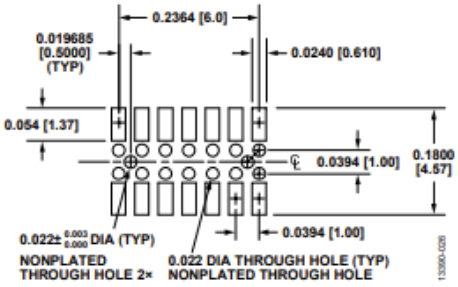


Figure 33. Mating Connector Design Detail

POWER SUPPLY CONSIDERATIONS

During startup, the internal power conversion system starts drawing current when VDD reaches 1.6 V. The internal processor begins initializing when VDD is equal to 2.35 V. After the processor starts, VDD must reach 2.7 V within 128 ms. Also, make sure that the power supply drops below 1.6 V to ensure that the internal processor shuts down. Use at least 10 µF of capacitance across VDD and GND. Best results come from using high quality, multilayer ceramic capacitors, located as close to the ADIS16460 connector as is practical. Using this capacitor supports optimal noise performance in the sensors.

BREAKOUT BOARD

The ADIS16IMU4/PCBZ breakout board provides a ribbon cable interface for simple connection to an embedded processor development system. Figure 34 shows the electrical schematic, and Figure 35 shows a top view for this breakout board. J2 mates directly to the electrical connector on the ADIS16460, and J1 easily mates to a 1 mm ribbon cable system.

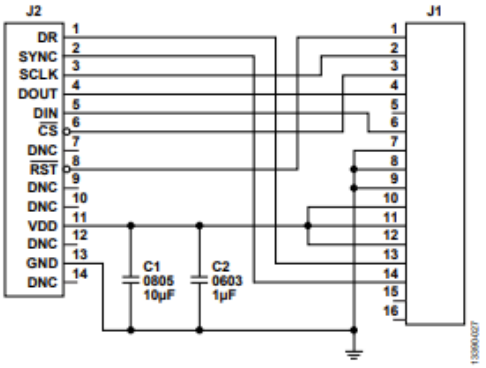


Figure 34. ADIS16IMU4/PCBZ Electrical Schematic

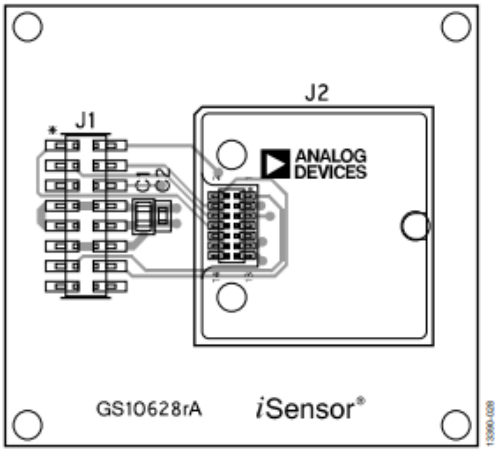


Figure 35. ADIS16IMU4/PCBZ Top View

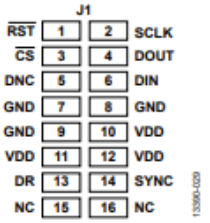


Figure 36. ADIS16IMU4/PCBZ J1 Pin Assignments

E.2. INA219

E.2.1 INA219 Application

9 Application and Implementation

NOTE

Information in the following applications sections is not part of the TI component specification, and TI does not warrant its accuracy or completeness. TI's customers are responsible for determining suitability of components for their purposes. Customers should validate and test their design implementation to confirm system functionality.

9.1 Application Information

The INA219 is a current shunt and power monitor with an I²C- and SMBus-compatible interface. The device monitors both a shunt voltage drop and bus supply voltage. Programmable calibration value, combined with an internal multiplier, enable readouts of current and power.

9.2 Typical Application

Figure 28 shows a typical application circuit for the INA219. Use a 0.1-µF ceramic capacitor for power-supply bypassing, placed as closely as possible to the supply and ground pins.

The input filter circuit consisting of R_{F1}, R_{F2}, and C_F is not necessary in most applications. If the need for filtering is unknown, reserve board space for the components and install 0-Ω resistors for R_{F1} and R_{F2} and leave C_F unpopulated, unless a filter is needed (see *Filtering and Input Considerations*).

The pull-up resistors shown on the SDA and SCL lines are not needed if there are pullup resistors on these same lines elsewhere in the system. Resistor values shown are typical: consult either the I²C or SMBus specification to determine the acceptable minimum or maximum values and also refer to the *Specifications* for Output Current Limitations.

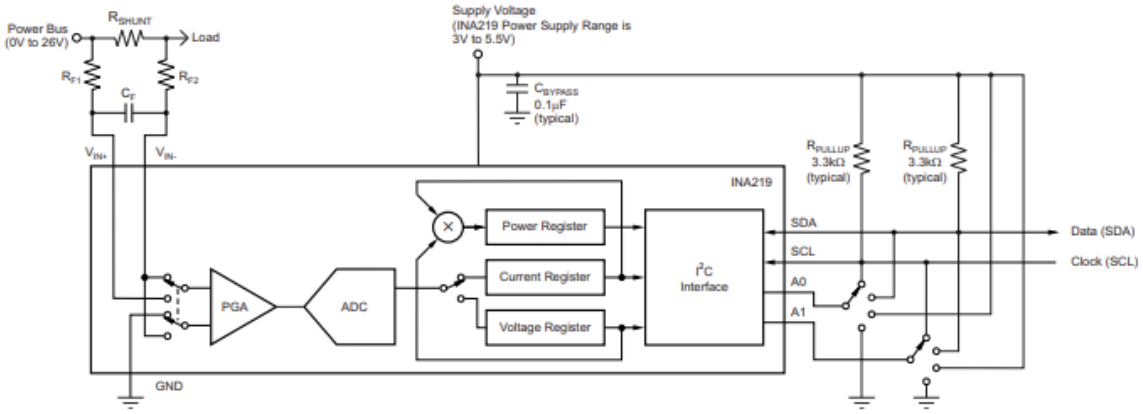


Figure 28. Typical Application Circuit

9.2.1 Design Requirements

The INA219 measures the voltage across a current-sensing resistor (R_{SHUNT}) when current passes through the resistor. The device also measures the bus supply voltage, and calculates power when calibrated. This section goes through the steps to program the device for power measurements, and shows the register results Table 8.

The Conditions for the example circuit is: Maximum expected load current = 15 A, Nominal load current = 10 A, V_{CM} = 12 V, R_{SHUNT} = 2 mΩ, V_{SHUNT} FSR = 40 mV (PGA = /1), and BRNG = 0 (VBUS range = 16 V).

9.2.2 Detailed Design Procedure

Figure 29 shows a nominal 10-A load that creates a differential voltage of 20 mV across a 2-mΩ shunt resistor. The common mode is at 12 volts and the voltage present at the IN– pin is equal to the common-mode voltage minus the differential drop across the resistor.

6.0 APPLICATIONS INFORMATION

6.1 Layout Considerations

The MCP9808 does not require any additional components besides the master controller in order to measure temperature. However, it is recommended that a decoupling capacitor of 0.1 µF to 1 µF be used between the V_{DD} and GND pins. A high-frequency ceramic capacitor is recommended. It is necessary for the capacitor to be located as close as possible to the power and ground pins of the device in order to provide effective noise protection.

In addition, good PCB layout is key for better thermal conduction from the PCB temperature to the sensor die. For good temperature sensitivity, add a ground layer under the device pins, as shown in Figure 6-1.

6.2 Thermal Considerations

A potential for self-heating errors can exist if the MCP9808 SDA, SCL and Event lines are heavily loaded with pull-ups (high current). Typically, the self-heating error is negligible because of the relatively small current consumption of the MCP9808. A temperature accuracy error of approximately +0.5°C could result from self-heating if the communication pins sink/source the maximum current specified.

For example, if the event output is loaded to maximum I_{OL}, Equation 6-1 can be used to determine the effect of self-heating.

EQUATION 6-1: EFFECT OF SELF-HEATING

$$T_\Delta = \theta_{JA}(V_{DD} \bullet I_{DD} + V_{OL\_Alert} \bullet I_{OL\_Alert} + V_{OL\_SDA} \bullet I_{OL\_SDA})$$

Where:

T_Δ = $T_J - T_A$
T_J = Junction Temperature
T_A = Ambient Temperature
θ_{JA} = Package Thermal Resistance
$V_{OL\_Alert, SDA}$ = Alert and SDA Output V_{OL} ($0.4\ V_{max}$)
$I_{OL\_Alert, SDA}$ = Alert and SDA Output I_{OL} ($3\ mA_{max}$)

At room temperature (T_A = +25°C) with maximum I_{DD} = 500 µA and V_{DD} = 3.6V, the self-heating due to power dissipation T_Δ is +0.2°C for the DFN-8 package and +0.5°C for the TSSOP-8 package.

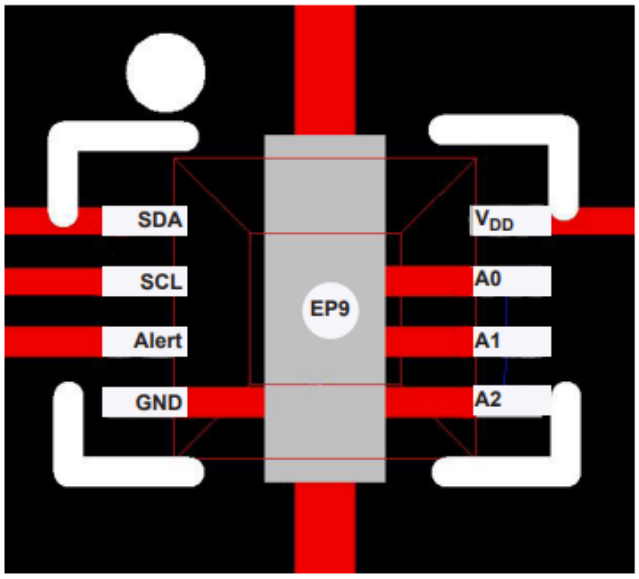
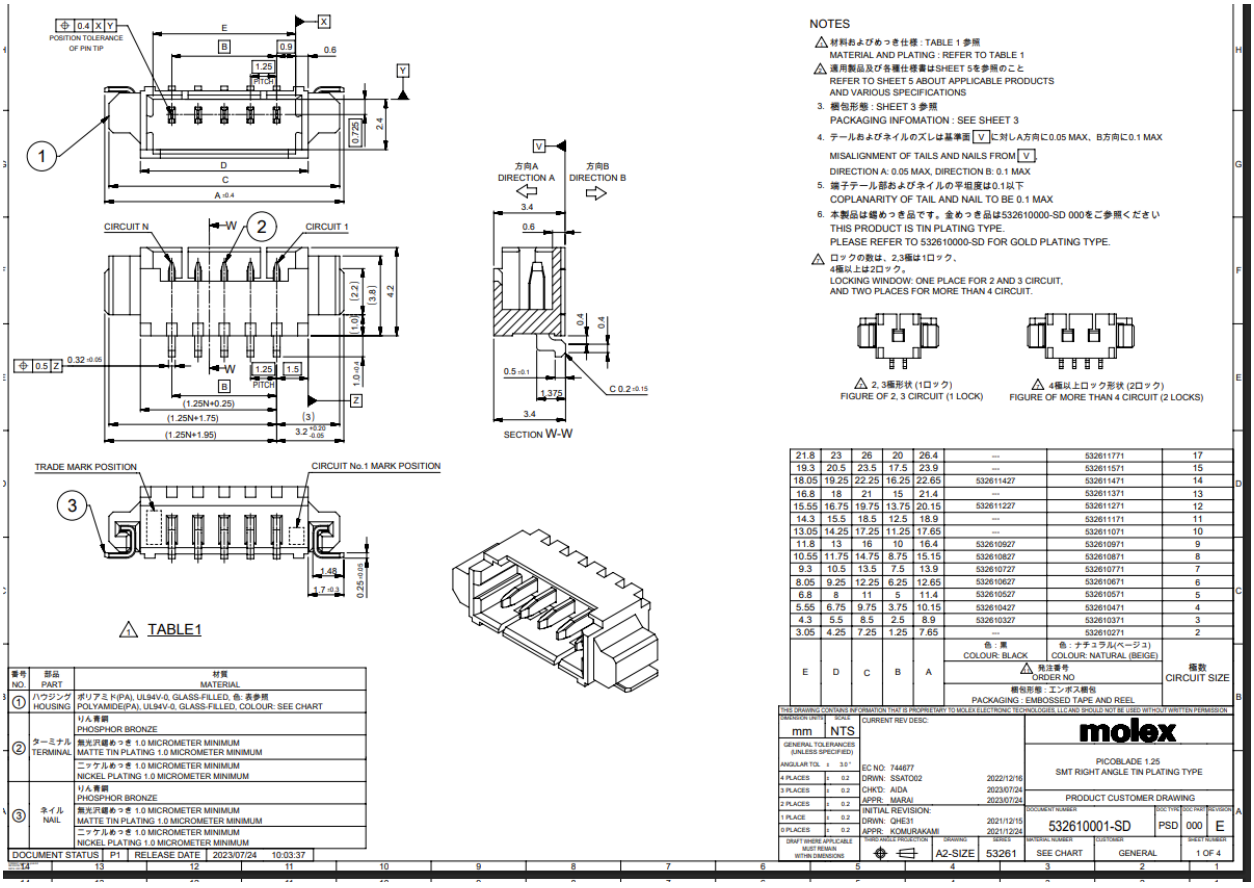


FIGURE 6-1: *DFN Package Layout (Top View).*

E.4. Connectors

E.4.1 Motor Connectors



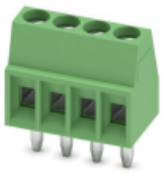
E.4.2. Power Supply Connector

MPT 0,5/ 4-2,54 - PCB terminal block

1725672

https://www.phoenixcontact.com/us/products/1725672

Please be informed that the data shown in this PDF document is generated from our online catalog. Please find the complete data in the user documentation. Our general terms of use for downloads are valid.



Printed circuit board terminal, nominal current: 6 A, rated voltage (III/2): 160 V, nominal cross section: 0.5 mm^2, number of potentials: 4, number of rows: 1, number of positions per row: 4, product range: MPT 0,5, pitch: 2.54 mm, connection method: Screw connection with tension sleeve, screw head form: L Slotted, mounting: Wave soldering, conductor/PCB connection direction: 0 °, color: green, Pin layout: Linear pinning, Solder pin [P]: 3.5 mm, number of solder pins per potential: 1, type of packaging: packed in cardboard

Your advantages

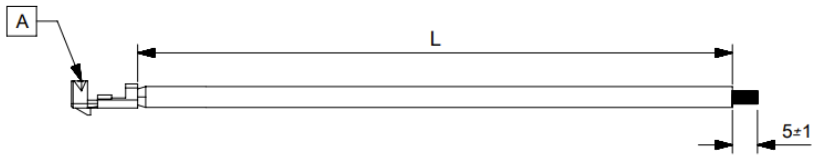
- Well-known connection principle allows worldwide use
- Low temperature rise, thanks to maximum contact force
- Allows connection of two conductors
- Extremely small design for the respective conductor cross section

Commercial data

Item number	1725672
Packing unit	250 pc
Minimum order quantity	250 pc
Sales key	AA11
Product key	AAKFAA
Catalog page	Page 83 (C-1-2013)
GTIN	4017918116279
Weight per piece (including packing)	1.212 g
Weight per piece (excluding packing)	1.2 g
Customs tariff number	85369010
Country of origin	GR

E.5. AWG Wire Harness

PART NUMBER	TITLE	CABLE DESCRIPTION	L	CABLE COLOR
214921-1121	PICOBLADE PRECRIMP 28AWG RD F-S L=75 Sn	1X28AWG DISC PVC RD UL1061 HOOKUP	75±3	RED
214921-1122	PICOBLADE PRECRIMP 28AWG RD F-S L=150 Sn	1X28AWG DISC PVC RD UL1061 HOOKUP	150±5	RED
214921-1123	PICOBLADE PRECRIMP 28AWG RD F-S L=225 Sn	1X28AWG DISC PVC RD UL1061 HOOKUP	225±5	RED
214921-1124	PICOBLADE PRECRIMP 28AWG RD F-S L=300 Sn	1X28AWG DISC PVC RD UL1061 HOOKUP	300±5	RED
214921-1125	PICOBLADE PRECRIMP 28AWG RD F-S L=450 Sn	1X28AWG DISC PVC RD UL1061 HOOKUP	450±5	RED
214921-1111	PICOBLADE PRECRIMP 28AWG BK F-S L=75 Sn	1X28AWG DISC PVC BK UL1061 HOOKUP	75±3	BLACK
214921-1112	PICOBLADE PRECRIMP 28AWG BK F-S L=150 Sn	1X28AWG DISC PVC BK UL1061 HOOKUP	150±5	BLACK
214921-1113	PICOBLADE PRECRIMP 28AWG BK F-S L=225 Sn	1X28AWG DISC PVC BK UL1061 HOOKUP	225±5	BLACK
214921-1114	PICOBLADE PRECRIMP 28AWG BK F-S L=300 Sn	1X28AWG DISC PVC BK UL1061 HOOKUP	300±5	BLACK
214921-1115	PICOBLADE PRECRIMP 28AWG BK F-S L=450 Sn	1X28AWG DISC PVC BK UL1061 HOOKUP	450±5	BLACK



NOTES:
1. Crimping to be done as per Molex standards.
2. The product specification relies on the terminals & wire spec.
3. Free ends must be stripped, twisted & tinned. The length of tinning is 5±1mm. Tinning Dia must be 0.46mm Max.

ITEM	PART#	DESCRIPTION	QTY	UOM
A	500798000	PicoBlade1.25 Crimp Rec Term Chain	1	PC

REVISION			
DATE	REV	DESCRIPTION	CHANGER
12/30/2019	A	INITIAL RELEASE	ARAJU
02/04/2020	A1	TINNING DIA CHANGED	ARAJU

DOCUMENT STATUS: P1 | RELEASE DATE: 2020/02/10 05:11:36

DIMENSION UNITS: mm | SCALE: NTS
GENERAL TOLERANCES (UNLESS SPECIFIED)
ANGULAR TOL ± °
4 PLACES ±
3 PLACES ±
2 PLACES ±
1 PLACE ±
0 PLACES ±
DRAFT WHERE APPLICABLE MUST REMAIN WITHIN DIMENSIONS

CURRENT REV DESC:
EC NO: 631848
DRWN: ARAJU 2020/02/04
CHK'D: RDESAI01 2020/02/10
APPR: RDESAI01 2020/02/10
INITIAL REVISION:
DRWN: ARAJU 2019/12/11
APPR: RDESAI01 2019/12/30

THIRD ANGLE PROJECTION | DRAWING: A4-SIZE | SERIES: 214921

molex

SEE TABLE

PRODUCT CUSTOMER DRAWING

DOCUMENT NUMBER	DOC TYPE	DOC PART	REVISION
2149211121	PSD	000	A1

MATERIAL NUMBER	CUSTOMER	SHEET NUMBER
SEE TABLE	GENERAL MARKET	1 OF 1

Ambient Light Sensor IC Series

Digital 16bit Serial Output Type Ambient Light Sensor IC

BH1730FVC

General Description

BH1730FVC is a digital Ambient Light Sensor IC with I²C bus interface. This IC is most suitable for obtaining ambient light data for adjusting LCD and backlight power of TV and mobile phone. It is capable of detecting a very wide range of illuminance.

Features

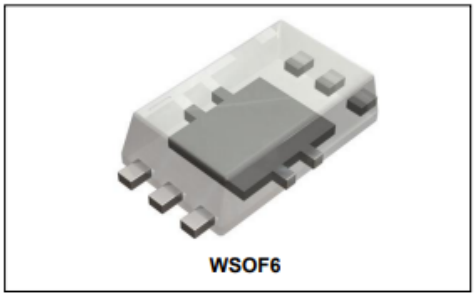
- I²C bus Interface
  f/s Mode Support, Slave Address "0101001"
- 2 outputs with peak wavelengths of visible light and infrared light respectively.
- Illuminance to digital converter
- Low current by power down function
- 50Hz / 60Hz light noise reject function
- Light source dependency is small by the calculation using 2 outputs. (e.g. Incandescent lamp, Fluorescent lamp, Halogen lamp, White LED and Sun light)
- Built-in interrupt function
- Sensitivity adjustment function for compensation for illuminance decrease by optical window

Key Specifications

- Supply Voltage Range: 2.4V to 3.6V
- I²C I/O Voltage: 1.65V to V_{CC} V
- Detection Range: 0.001 lx to 100k lx
- Current Consumption: 150 µA (Typ)
- Power Down Current: 0.85 µA (Typ)
- Operating Temperature Range: -40°C to +85°C

Package

Package	W(Typ) x D(Typ) x H(Max)
WSOF6	1.60mm x 3.00mm x 0.75mm

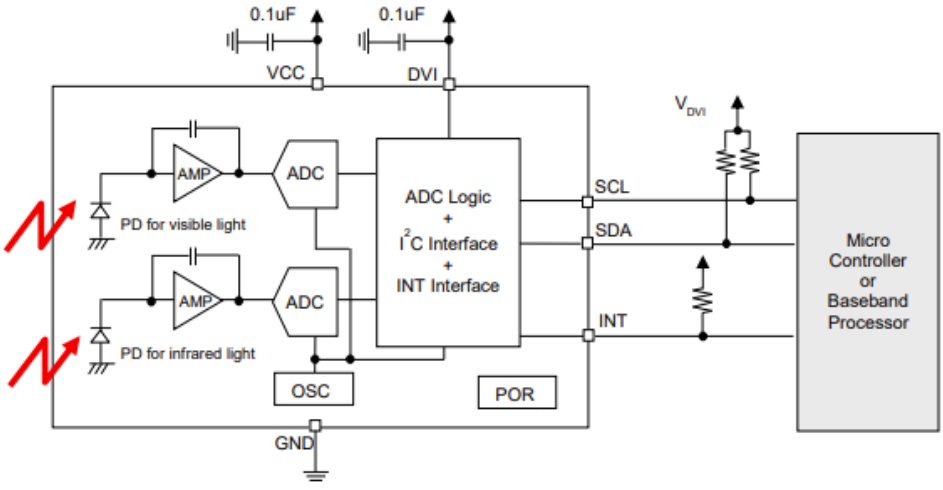


WSOF6

Applications

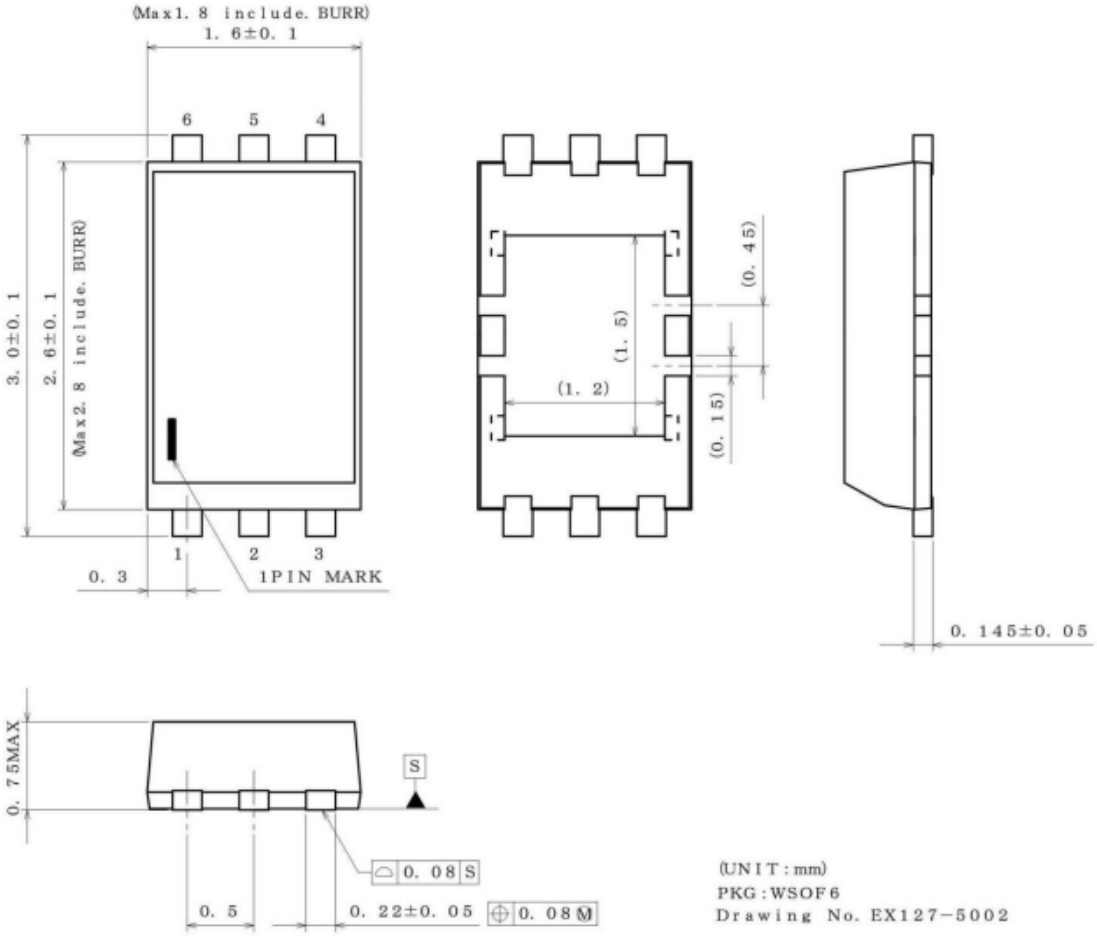
LCD TV, Mobile Phone, Tablet PC, Note PC,
Digital Camera, Portable Game Machine

Typical Application Circuit



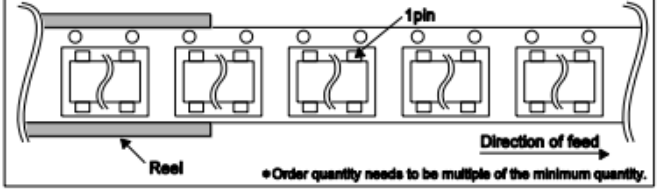
Physical Dimension, Tape and Reel Information

Package Name	WSOF6



<Tape and Reel Information>

Tape	Embossed carrier tape
Quantity	3000pcs
Direction of feed	TR (The direction is the 1pin of product is at the upper right when you hold reel on the left hand and you pull out the tape on the right hand)



E.7 Solder Data

285 Flux-Cored Wire

Mildly Activated Rosin Cored Wire for Leaded and Lead-free Alloys

Product Description

Kester 285 Flux-Cored Wire is a mildly activated rosin flux is classified as Type ROL0 flux under IPC J-STD-004. This flux was formerly classified as Type RMA per QQ-S-571. 285 consists of high quality, purified rosin to which a synergistic combination of activating agents has been incorporated. The fluxing ability of 285 is much greater than ordinary mildly activated rosin fluxes and is comparable to fully activated rosin fluxes. 285 has been developed for use in the electronic industry where difficult assemblies are to be soldered, but process requirements stipulate use of a mildly activated rosin flux.

Performance Characteristics:

- Industry standard RMA cored wire
- Compatible with leaded and lead-free alloys
- Classified as ROL0 per J-STD-004

RoHS Compliance

This product meets the requirements of the Restriction of Hazardous Substances (RoHS) Directive, 2011/65/EU for the stated banned substances. (Applies only if this core flux is combined with a lead-free alloy.)

Reliability Properties

Copper Mirror Corrosion: Low
Tested to J-STD-004, IPC-TM-650, Method 2.3.32

Corrosion Test: Low
Tested to J-STD-004, IPC-TM-650, Method 2.6.15

Silver Chromate: Pass
Tested to J-STD-004, IPC-TM-650, Method 2.3.33

Chloride and Bromides: None Detected
Tested to J-STD-004, IPC-TM-650, Method 2.3.35

Fluorides by Spot Test: Pass
Tested to J-STD-004, IPC-TM-650, Method 2.3.35.1

E.8 Solder Flux Data

CHIPQUIK® SMD291NL

Datasheet revision 1.4 www.chipquik.com

Tack Flux No-Clean (for Lead-Free applications) in 5cc Syringe

Product Highlights

Ideal for all rework, solder, de-solder and reflow applications
Non-corrosive, non-conductive, no-clean
Tack flux will not run all over PCB when applied
Has a pleasant odor
Excellent wetting
Easily cleaned with isopropyl alcohol (IPA)
Attachment of BGA spheres
Soldering flip chip components
Long stencil life
Wide process window
Clear residue
For Lead-Free applications
RoHS 3 and REACH compliant

Representative Photo Only
Please note flux color may vary from light yellow to orange/brown

Specifications

Flux Type:	Synthetic No-Clean (for Lead-Free applications)
Flux Classification:	REL0
Flux Activation Temperature:	140°C (284°F)
Color:	Light yellow to orange/brown
Packaging:	5cc/5g Syringe
Shelf Life:	Refrigerated >24 months, Unrefrigerated >24 months

Stencil Life

>8 hours @ 20-50% RH 22-28°C (72-82°F)
>4 hours @ 50-70% RH 22-28°C (72-82°F)

Stencil Cleaning

Automated stencil cleaning systems for both stencil and misprinted boards. Manual cleaning using isopropyl alcohol (IPA).

Storage and Handling

Store refrigerated or at room temperature 3-25°C (37-77°F). Do not freeze. Allow 4 hours for flux to reach an operating temperature of 20-25°C (68-77°F) before use.

Transportation

This product has no shipping restrictions. Shipping below 0°C (32°F) or above 25°C (77°F) for normal transit times by ground or air will not impact this product's stated shelf life.

Conforms to the following Industry Standards:

J-STD-004B, Amendment 1 (Solder Fluxes):	Yes
RoHS 3 Directive (EU) 2015/863:	Yes